For the sensor you need a raspberry pi connected to a DHT11 temperature and humidity sensor. To install the sensor connect the power(+) wire to pin-2, the ground(-) wire to pin-6, and the data wire to pin-7 which is GPIO4. Other wire set ups can work depending on the configuration of your pi’s pins, but changing where the data wire is connected requires a small adjustment to the ‘temperatureAndHumidityRead.py’ code. This adjustment can be made by modifying the line “dhtDevice = Adafruit\_dht.DHT11(board.D4)” and changing “4” to the number of the GPIO port you wish to use. Note that the number of the GPIO does not necessarily match the number of the pin.

To run the ‘tempuratureAndHumidityRead.py’ program you will need to download certain supporting software to the device. The three key libraries needed are Adafruit-blinka, Adafruit-CircuitPython-BusDevice, and Adafruit-circuitpython-dht. Assuming your raspberry pi is using a Linux operating system with Python pre-installed these can be downloaded by simply opening the terminal and typing “pip install <insert name of library>”. These libraries are essential as they enable the program to interface with the hardware of the DHT11 sensor. If Python/pip are not already set up on your device, you can add them from the terminal by using the commands “sudo apt-get update”, “sudo apt-get install python3”, and then “sudo apt-get install python3-pip”.

Once these libraries are successfully installed there is one last step before the program can be run. Open the ‘temperatureAndHumidityRead.py’ file and look for the line labeled with “#Server info”. In the line immediately below you must adjust the URL to reflect the IP address where you are running the application backend program ‘App\_Mongo.js’. At this point the code is ready to run by using the ‘run’ button in the editor or running it from the terminal, but you can also take this moment to make desired adjustments to the function if desired. The code is set up to be easily adjustable for key components. By default, the code is set to collect data and post it to the database once every 30 minutes, but by changing the value of ‘waitTime’ you can set it to run as often as you want by setting waitTime equal to the desired timespan in seconds. However it is important to note that running more than once per 2 seconds can cause issues with collecting accurate data.

In order for data to be added or retrieved from the database the backend server must be running. The server only requires the user device to have node.js installed, it can then be run with the terminal command ‘node App\_Mongo.js’. This program will handle requests made to the MongoDB database. If you wish to change the database these programs interface with then you must change the variable ‘dbName’ to reflect the name of the new database.