

PUBLISH DATA TO THE IBM CLOUD

Team ID	PNT2022TMID05259
Project Name	Smart Waste Management System For Metropolitan cities

AIM:

To send sensor data (or any dummy data) from Raspberry –Pi to IBM Watson. In our case it is DHT Sensors Data.

REQUIREMENTS:

HARDWARE:

- RASPBERRY-PI (3B)(WITH ETHERNET CABLE OR WIFI CONNECTED)
- USB MOUSE
- USB KEYBOARD
- VGA TO HDMI CABLE
- A MONITOR
- RASPBERRY’S POWER SUPPLY
- DHT-11 Sensor ○ Connecting Wires

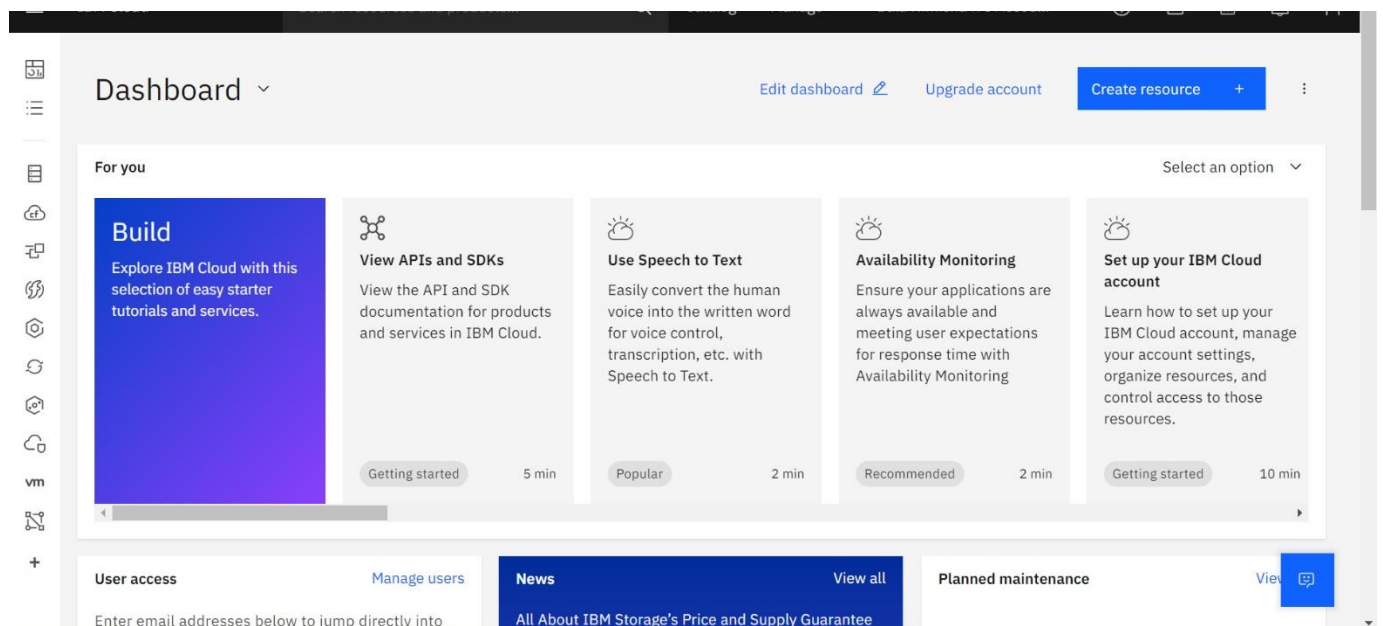
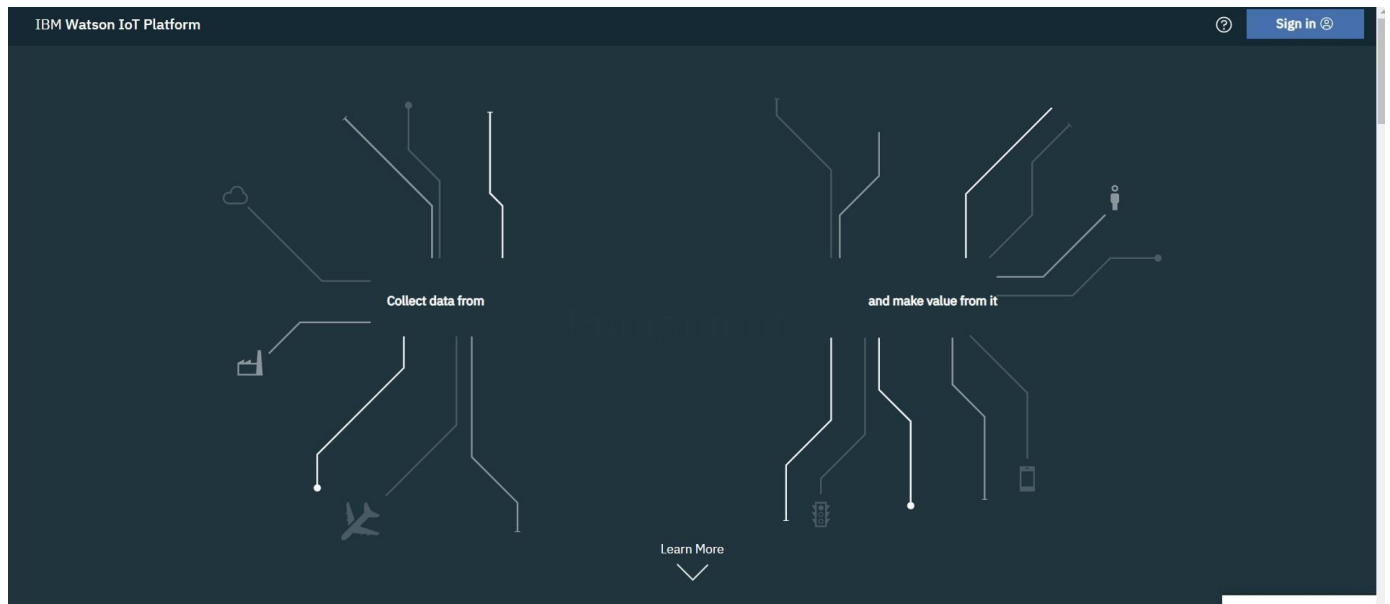
SOFTWARE:

- IBM BLUEMIX ACCOUNT

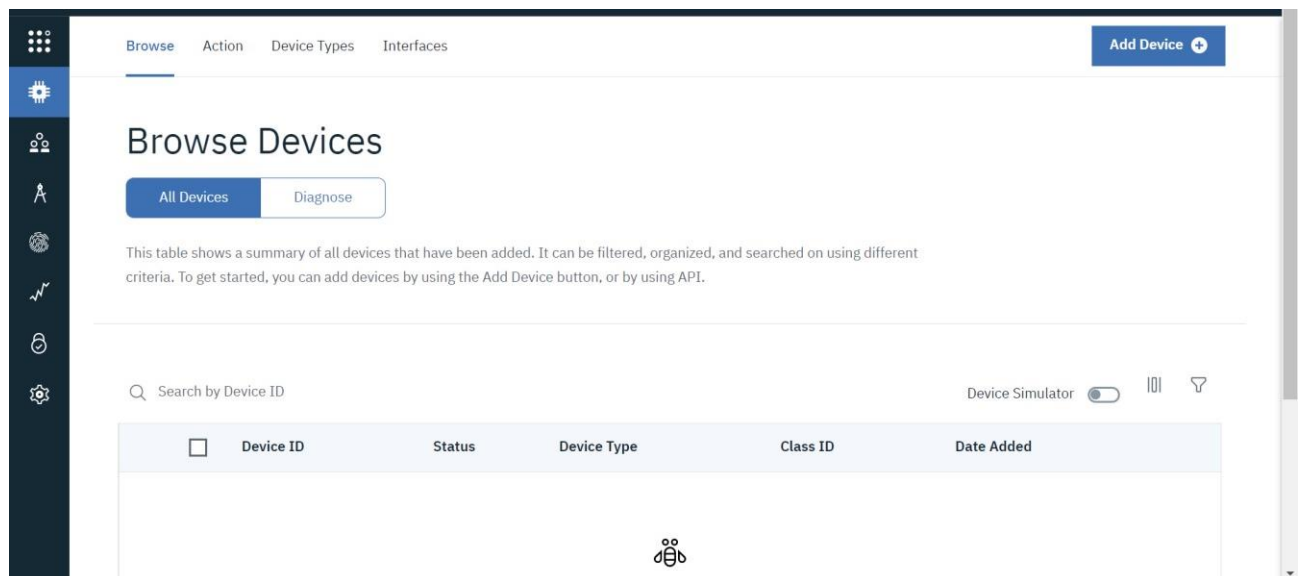
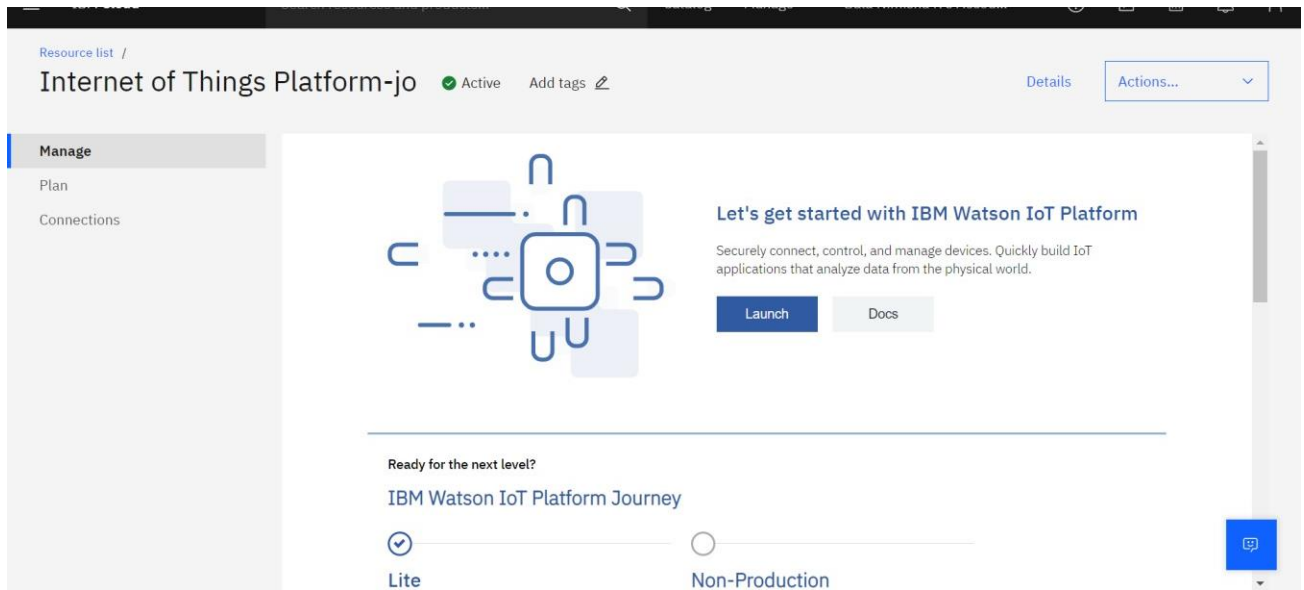
STEPS TO BE FOLLOWED

Step-1: Create a device in IBM Watson:

- Firstly, login into your IBM-Bluemix account with your e-mail ID and Password.



Check all details and click create



- After click on Add device this page will open

Browse Action Device Types Interfaces

Add Device

Identity Device Information Security Summary

Select a device type for the device that you are adding and give the device a unique ID.

Device Type

Device ID

Cancel Next

- Go to device type and fill the details.

Browse Action Device Types Interfaces

Add Type

Identity Device Information

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

Type Or

Name

The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.

Description

Cancel Next

- Click on Finish

Browse Action Device Types Interfaces

Register Devices, Define Interfaces

Now that you added a device type, you can register and connect devices for this type.

Cancel Next

- Choose the device and give device ID and then click on next.

The screenshot shows a web application interface with a dark sidebar on the left containing icons for various functions. The main content area has a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. Below this, a light blue box contains the text 'Select a device type for the device that you are adding and give the device a unique ID.' There are two input fields: 'Device Type' with the value 'Fantastic-4' and 'Device ID' with the value '12345'. At the bottom right of this box are 'Cancel' and 'Next' buttons. Below the light blue box, the heading 'Browse Devices' is followed by two buttons: 'All Devices' and 'Diagnose'.

- Click on Next

The screenshot shows a web application interface with a dark sidebar on the left. The main content area has a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. Below this, a light blue box contains the text 'You can modify the default device information and enter more information about the device for identification purposes.' There are several input fields: 'Serial Number', 'Model', 'Description', 'Hardware Version', 'Manufacturer', 'Device Class', 'Firmware Version', and 'Descriptive Location'. Each field has a placeholder text 'Enter [field name]'. Below these fields is a button labeled 'Add Metadata' with a plus icon. At the bottom right of the light blue box are 'Back' and 'Next' buttons.

The screenshot shows a web application interface with a dark sidebar on the left. The main content area has a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. Below this, the text 'There are two options for selecting a device authentication token.' is followed by two columns. The left column is titled 'Auto-generated authentication token (default)' and contains the text 'Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphanumeric characters and symbols. The token is returned to you at the end of the device registration process.' The right column is titled 'Self-provided authentication token' and contains the text 'Provide your own authentication token for this device. The token must be between 8 and 36 characters and contain a mix of lowercase and uppercase letters, numbers, and symbols, which can include hyphens, underscores, and periods. Do not use repeated characters, dictionary words, user names, or other predefined sequences.' Below these columns is an input field labeled 'Authentication Token' with the placeholder text 'Enter an optional token' and an information icon. Below the input field is the text 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' and 'Authentication token are encrypted before we store them.'

The screenshot displays the IBM Watson IoT Platform interface. On the left is a dark sidebar with icons for navigation. The main content area has a top navigation bar with tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue button labeled 'Add Device' is in the top right corner. Below the navigation bar, a header bar for a specific device shows the ID '12345', status 'Disconnected', name 'Fantastic-4', and a timestamp 'Nov 7, 2022 10:15 AM'. Below this header are five tabs: 'Identity', 'Device Information', 'Recent Events' (which is selected), 'State', and 'Logs'. A text message states: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this text is a table with the following data:

Event	Value	Format	Last Received
event_1	{"type":"Buffer","data":[]}	json	a few seconds ago
event_1	{"#IBM Watson IOT Platform":"#pip install wiot..."}	json	5 minutes ago
event_1	{"randomNumber":24}	json	5 minutes ago

A small white tooltip box is visible at the bottom right of the screenshot, containing the text '1 Simulation running'.

STEP-2: INSTALLING NECESSARY PACKAGES ON YOUR PI:

- Now we are going to install necessary packages on your pi.
- Open your terminal in your pi and type the following commands
- `curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb`
- `sudo dpkg -i iot_1.0-2_armhf.deb`
- `service iot status`

Following are the images as to what appears on your pi's terminal when u type these commands

```
File Edit Tabs Help
--2017-10-23 08:55:22-- http://ftp.nl.debian.org/debian/pool/main/o/openssl/lib
s11.0.0-1.0-1-1-deb8u6_armhf.deb
Resolving ftp.nl.debian.org (ftp.nl.debian.org)... 130.89.149.21, 2001:67c:2564:
a120::21
Connecting to ftp.nl.debian.org (ftp.nl.debian.org)[130.89.149.21]:80... connect
ed
HTTP request sent, awaiting response... 200 OK
Length: 867950 (848K) [application/x-debian-package]
Saving to: 'libssl1.0.0-1.0-1-1-deb8u6_armhf.deb'

libssl1.0.0-1.0-1- 100%[=====] 847.61K 358KB/s in 2.4s

2017-10-23 08:55:25 (358 KB/s) - 'libssl1.0.0-1.0-1-1-deb8u6_armhf.deb' saved [
867950/867950]

pi@raspberrypi:~$ sudo dpkg -i libssl1.0.0-1.0-1-1-deb8u6_armhf.deb
Selecting previously unselected package libssl1.0.0:armhf.
(Reading database ... 115600 files and directories currently installed.)
Preparing to unpack libssl1.0.0-1.0-1-1-deb8u6_armhf.deb ...
Unpacking libssl1.0.0:armhf (1.0.0-1-1-deb8u6) ...
Setting up libssl1.0.0:armhf (1.0.0-1-1-deb8u6) ...
pi@raspberrypi:~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/rel
eases/download/1.0.2.1/iot_1.0-2_armhf.deb
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 164 0 164 0 0 157 0 --:--:-- 0:00:01 --:--:-- 157
100 600 0 600 0 0 457 0 --:--:-- 0:00:01 --:--:-- 457
100 110k 100 110k 0 0 20117 0 0:00:03 0:00:03 --:--:-- 48190
pi@raspberrypi:~$ sudo dpkg -i iot_1.0-2_armhf.deb
(Reading database ... 115626 files and directories currently installed.)
Preparing to unpack iot_1.0-2_armhf.deb ...
Unpacking iot (1.0-1) over (1.0-1) ...
Setting up iot (1.0-1) ...
Processing triggers for systemd (232-25+deb9u1) ...
pi@raspberrypi:~$ service iot status
* iot.service - LSB: IoT service
Loaded: loaded (/etc/init.d/init; generated; vendor preset: enabled)
Active: active (running) since Mon 2017-10-23 08:56:25 UTC; 17s ago
Docs: man:systemd-sysv-generator(8)
CGroup: /system.slice/iot.service
└─2562 /opt/iot/iot /dev/null

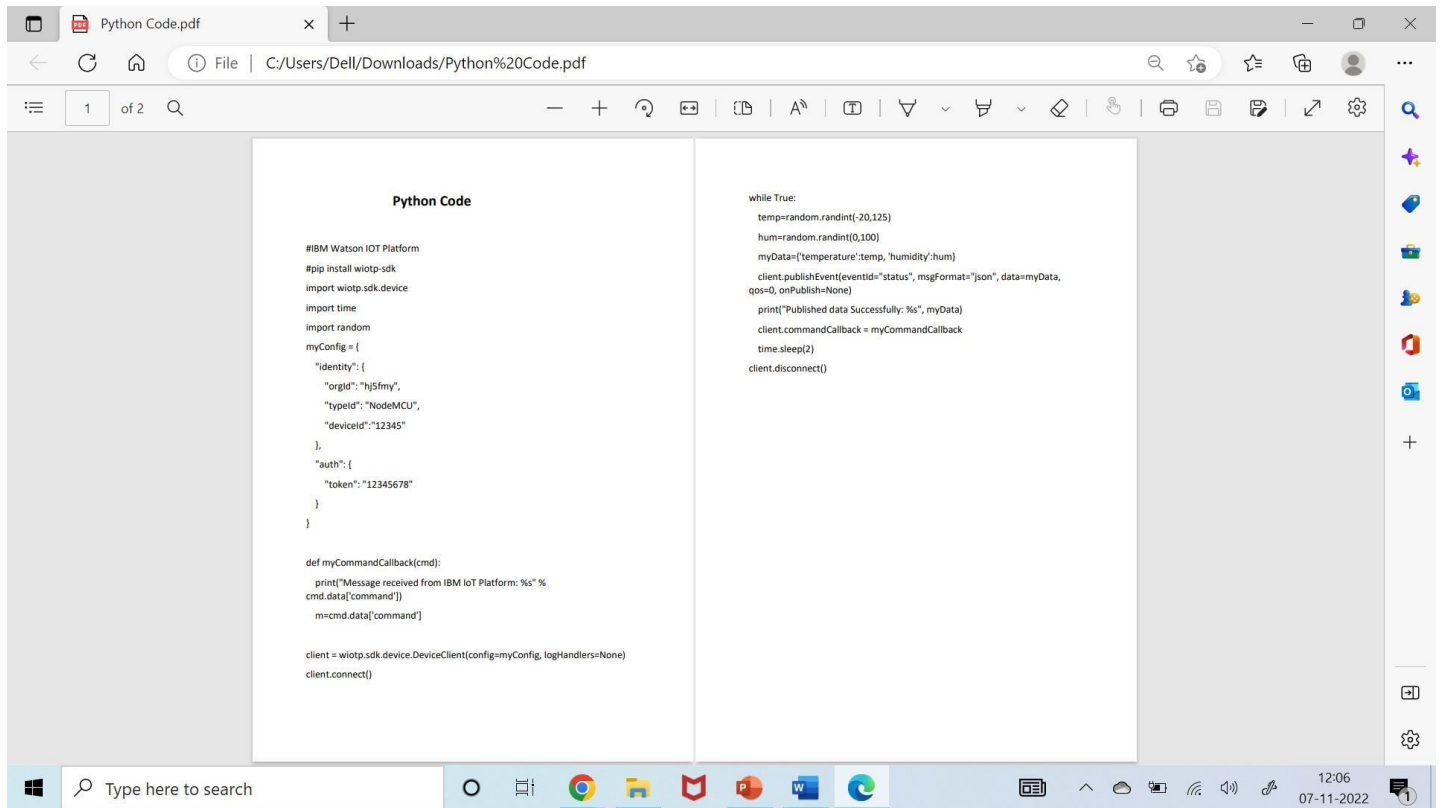
Oct 23 08:56:24 raspberrypi systemd[1]: Starting LSB: IoT service...
Oct 23 08:56:24 raspberrypi iot[2557]: Starting the iot program
Oct 23 08:56:25 raspberrypi iot[2562]: **** IoT Raspberry Pi Sample has started ****
Oct 23 08:56:25 raspberrypi iot[2562]: Config file not found. Going to Quickstart mode
Oct 23 08:56:25 raspberrypi iot[2562]: Running in Quickstart mode
Oct 23 08:56:25 raspberrypi systemd[1]: Started LSB: IoT service.
```

- Then open your terminal and type `pip install ibm-iotf`

```
File Edit Tabs Help
pi@raspberrypi:~$ pip install ibmiotf
Collecting ibmiotf
  Downloading ibmiotf-0.3.0.tar.gz (58kB)
    100% |#####| 61kB 510kB/s
Collecting dicttoxml>=1.7.4 (from ibmiotf)
  Downloading dicttoxml-1.7.4.tar.gz
Collecting iso8601>=0.1.10 (from ibmiotf)
  Downloading iso8601-0.1.12-py2.py3-none-any.whl
Collecting paho-mqtt>=1.2 (from ibmiotf)
  Downloading paho-mqtt-1.3.1.tar.gz (80kB)
    100% |#####| 81kB 916kB/s
Collecting pytz>=2014.7 (from ibmiotf)
  Using cached pytz-2017.2-py2.py3-none-any.whl
Collecting requests>=2.5.0 (from ibmiotf)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 1.6MB/s
Collecting requests-toolbelt>=0.7.0 (from ibmiotf)
  Downloading requests-toolbelt-0.8.0-py2.py3-none-any.whl (54kB)
    100% |#####| 61kB 1.6MB/s
Collecting xmldict>=0.10.2 (from ibmiotf)
  Downloading xmldict-0.11.0-py2.py3-none-any.whl
Collecting urllib3<1.23,=>1.21.1 (from requests>=2.5.0->ibmiotf)
  Downloading urllib3-1.22-py2.py3-none-any.whl (122kB)
    100% |#####| 133kB 1.4MB/s
Collecting idna<2.7,=>2.5 (from requests>=2.5.0->ibmiotf)
  Downloading idna-2.6-py2.py3-none-any.whl (58kB)
    100% |#####| 61kB 1.7MB/s
Collecting chardet<3.1.0,=>3.0.2 (from requests>=2.5.0->ibmiotf)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.6MB/s
Collecting certifi>=2017.4.17 (from requests>=2.5.0->ibmiotf)
  Using cached certifi-2017.7.27.1-py2.py3-none-any.whl
Building wheels for collected packages: ibmiotf, dicttoxml, paho-mqtt
Running setup.py bdist_wheel for ibmiotf ... done
Stored in directory: /home/pi/.cache/pip/wheels/7e/fa/45/bbc33ad957e82f7b71ba80e31d65a83d9d735a0d12e0c9418
Running setup.py bdist_wheel for dicttoxml ... done
Stored in directory: /home/pi/.cache/pip/wheels/45/62/59/96910b33ec6a7b2ae66a13765401b50def468024078e12cce
Running setup.py bdist_wheel for paho-mqtt ... done
Stored in directory: /home/pi/.cache/pip/wheels/20/d8/0d/acdc8f289011b7be7de71deebe642fb83be0313dfff0493
Successfully built ibmiotf dicttoxml paho-mqtt
Installing collected packages: dicttoxml, iso8601, paho-mqtt, pytz, urllib3, idna, chardet, certifi, requests, requests-toolbelt, xmldict, ibmiotf
Successfully installed certifi-2017.7.27.1 chardet-3.0.4 dicttoxml-1.7.4 ibmiotf-0.3.0 idna-2.6 iso8601-0.1.12 paho-mqtt-1.3.1 pytz-2017.2 requests-2.18.4 requests-toolbelt-0.8.0 urllib3-1.22 xmldict-0.11.0
pi@raspberrypi:~$
```

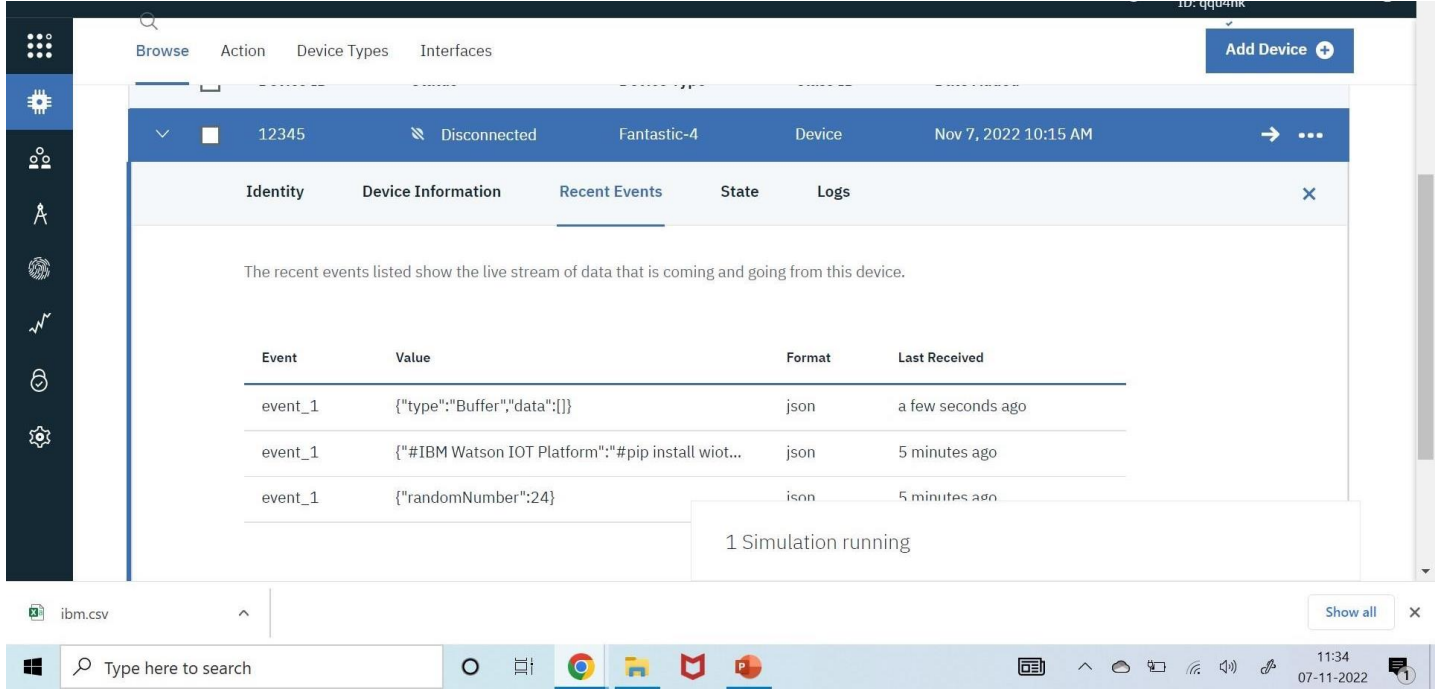
I have sent DHT-11 Sensors data to ibm bluemix .To get the code u need to login into IOT GYAN.

- Then I get the image as follows in my pi's shell:



Step-3: checking your data sent on IBM Bluemix:

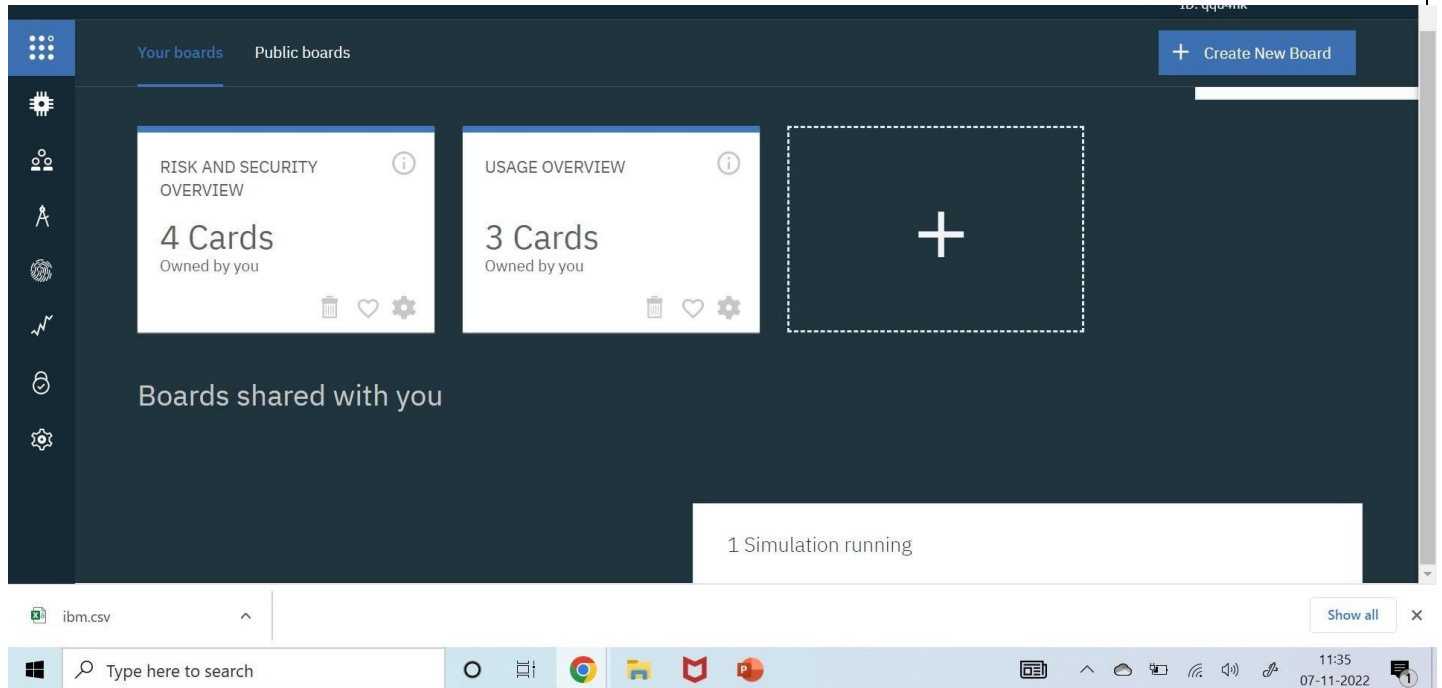
- After you have sent your sensors data you can check whether it is received at your iot platform Just look at the image below and if u see the same wifi kind of symbol on your created device then your data is being received.



After double clicking on your created device you can see the received data as shown in image

Step-4: Creating boards and cards for visualization of data:

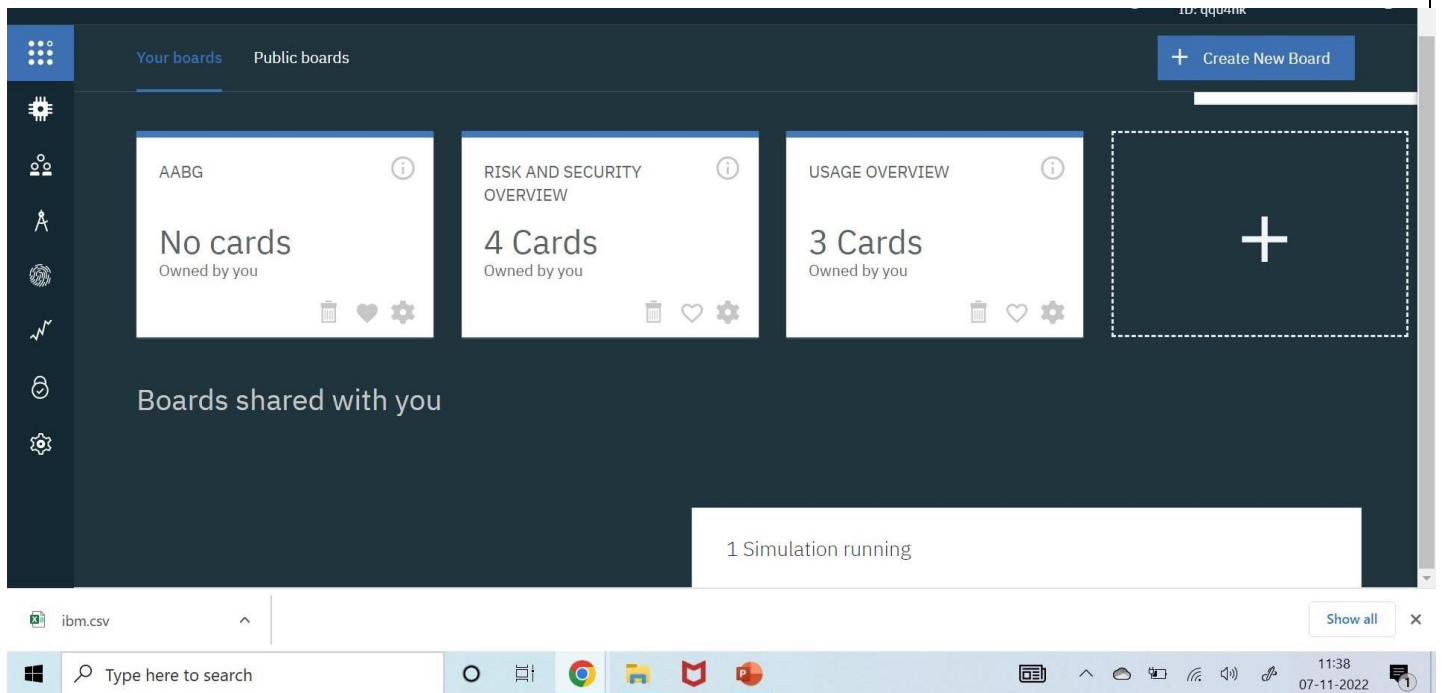
- In your Watson platform you have an option called board .Click on it and you get the following window on your screen



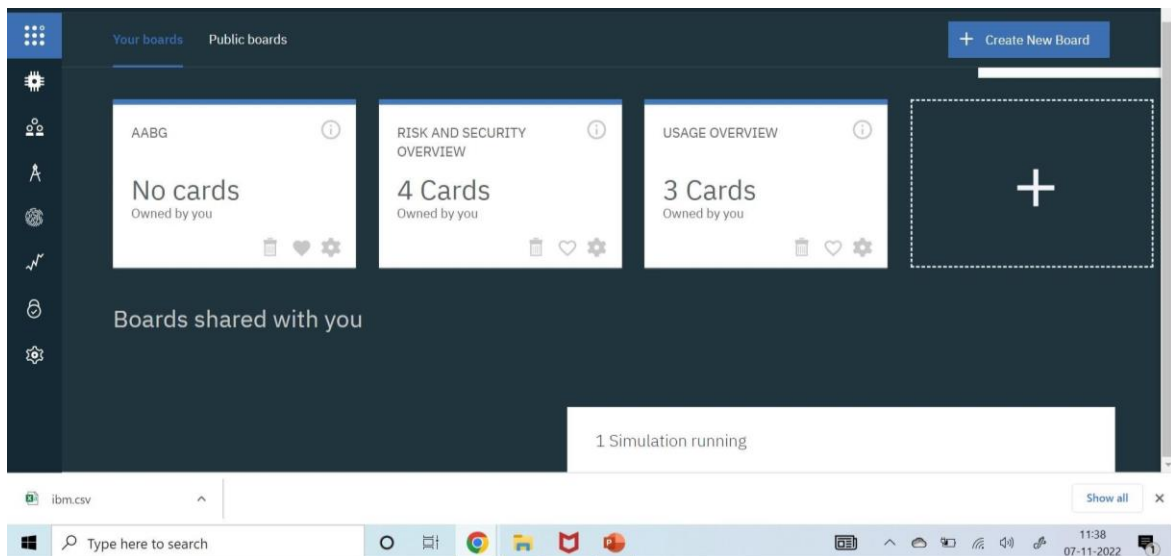
- Click on Create a new board to create a board .

The given below window appears give a name and description to your board as shown in the window below.

○

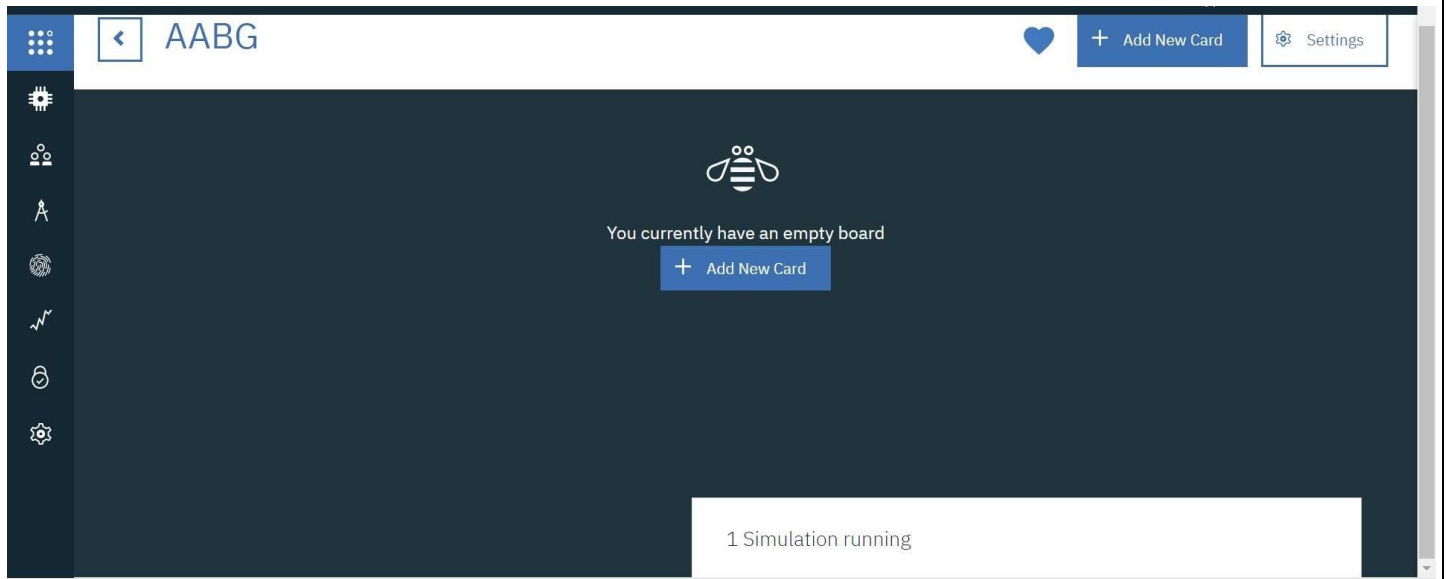


○ Then click on Next you get the below window then again click on Sub

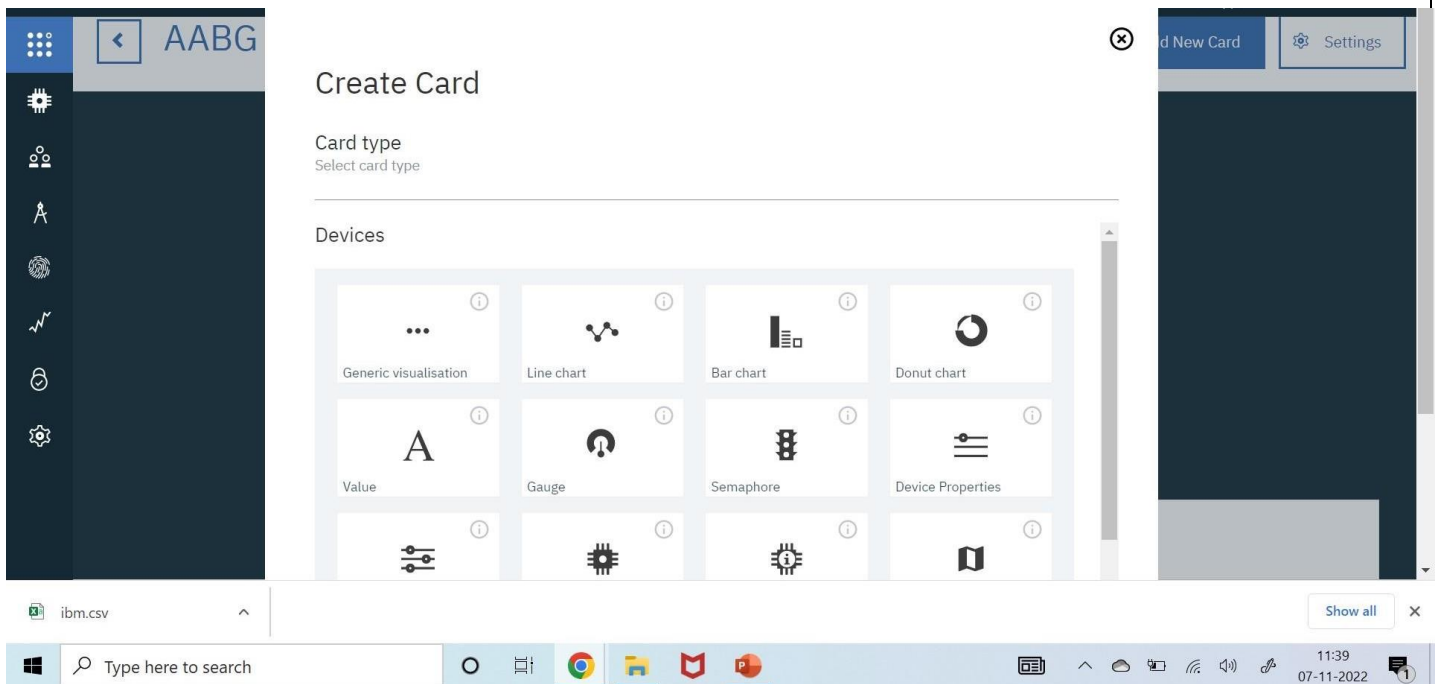


Then double click on your boards name which you have created.

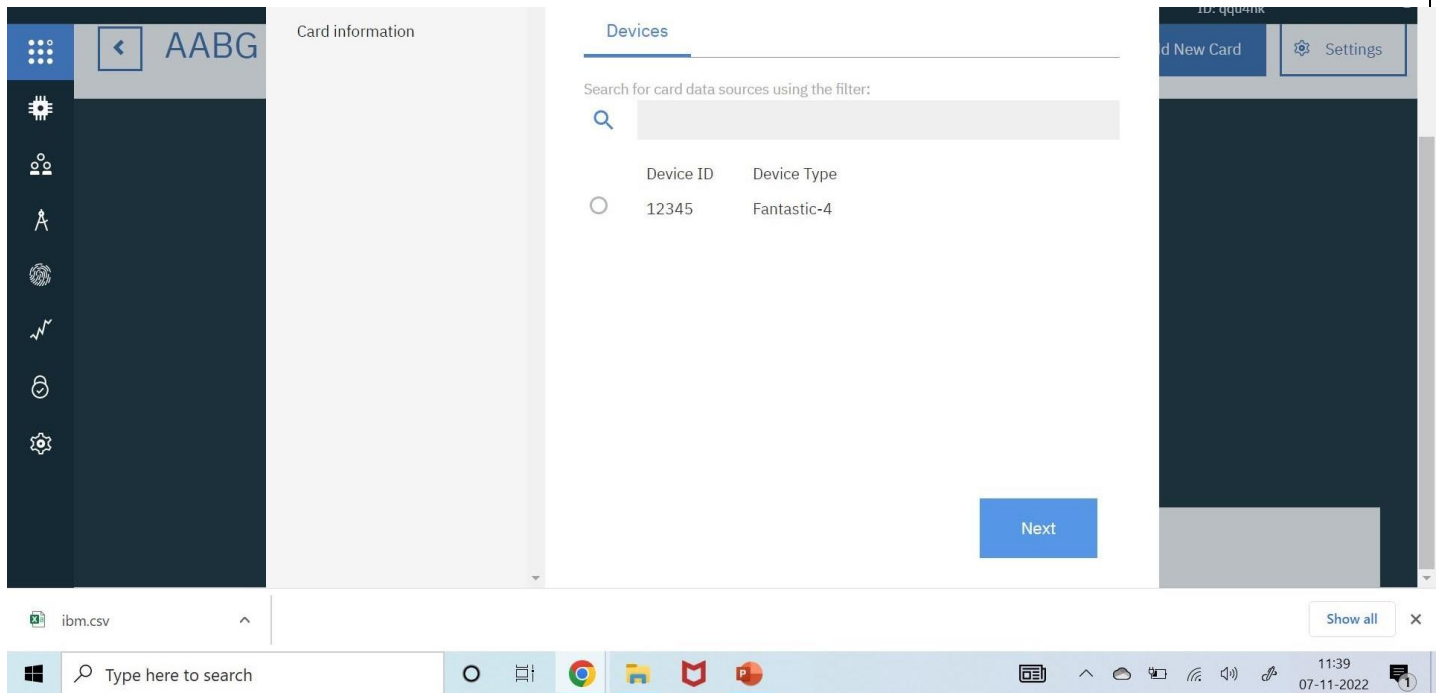
○ Click on Add New Card



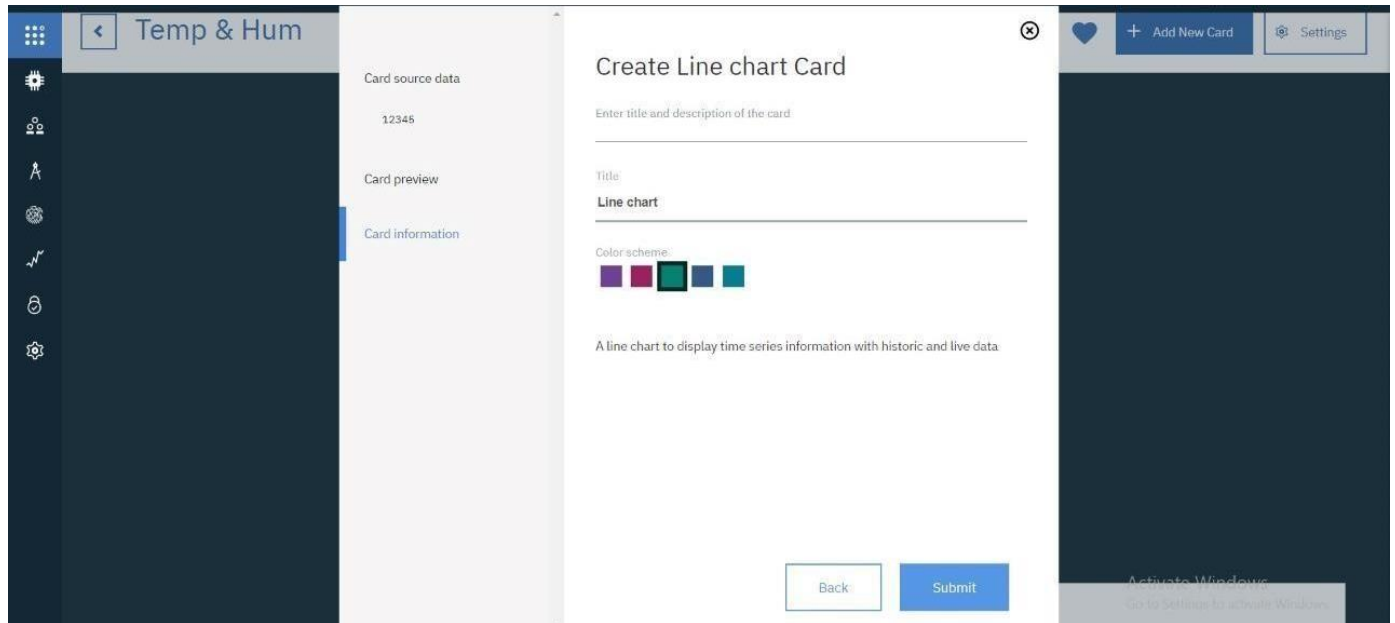
- Select the type of Graph u want accordingly and click next



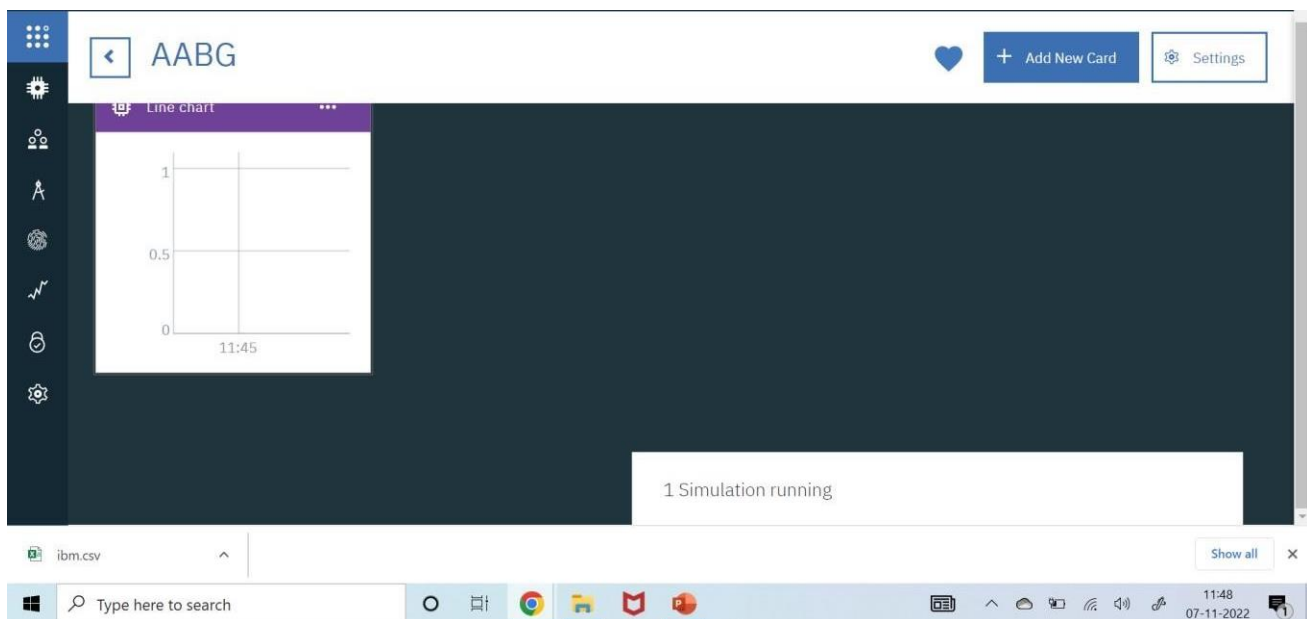
- You get the below window, choose the Device and click on Next.



- Select the event, properly to be visualized on your graph and click next. In my case it is humidity
- Then select the size of the graph and color of the graph board you want and click next



○ Here is the graph



○ Repeat the process to get different graphs.

RESULT:

Hence, we were able to send data from our pi to IBM Watson and visualize it on a graph.