**PROJECT REPORT**

**Title:**

Smart Agriculture System Based on IoT-SB50433

Internship at SmartInternz



**Application ID** - SPS\_APL\_20200004106

**Project ID** - SPS\_PRO\_101

**Name** – Snehasish Pradhan

**Email ID** – sailofthoughts@gmail.com

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | |  | **INTRODUCTION** | | -----------------**03** |  |  |  |
|  | |  | 1.1 Overview | |  |  |  |  |
|  | |  | 1.2 Purpose | |  |  |  |  |
| **2** | |  | **LITERATURE SURVEY** | | ------------------ **05** |  |  |  |
|  | |  | 2.1 Existing problem | | |  |  |  |
|  | |  | 2.2 Proposed solution | | |  |  |  |
| **3** | |  | **THEORITICAL ANALYSIS** | | -------------------**06** |  |  |  |
|  | |  | 3.1 Block diagram | |  |  |  |  |
|  | |  | 3.2 Hardware / Software designing | | |  |  |  |
| **4** | |  | **EXPERIMENTAL INVESTIGATIONS --------------- 07** | | |  |  |  |
| **5** | |  | **FLOWCHART** | | ------------------------ **09** |  |  |  |
| **6** | |  | **RESULT** |  | ---------------------- **11** |  |  |  |
| **7** | |  | **ADVANTAGES & DISADVANTAGES ----------------- 11** | | |  |  |  |
| **8** | |  | **APPLICATIONS** | | ------------------------- **12** |  |  |  |
| **9** | |  | **CONCLUSION** | | -------------------------- **13** |  |  |  |
| **10** | |  | **FUTURE SCOPE** | | **------------------- 13** |  |  |  |
| **11** | |  | **BIBILOGRAPHY** | | ------------------------- **13** |  |  |  |
|  | |  | **APPENDIX** | |  |  |  |  |
|  |  | | A. Source code | | ------------------------- **14** |  |  |  |

**Table of Contents**

**1. INTRODUCTION**

**1.1 Overview**

Agriculture is major source of income for the largest population in India and is major contributor to Indian economy. In past decade it is observed that there are not much crop development in agriculture sector. There are number of factor which is responsible for this it may be due to water waste, low soil fertility, Fertilizer abuse, climate change or diseases etc. It is very essential to make effective intervention in agriculture and the solution is IOT in integration with wireless sensor network. IoT is network that connects physical objects or things embedded with electronics, software and sensors through network connectivity that collects and transfers data using cloud for communication.

The project proposes to create a novel smart IOT based agriculture system which will assist farmers in getting live data (Temperature, soil moisture, humidity) for efficient environment monitoring which will enable them to do smart farming and increase their overall yield and quality of products.

In this project a web app is created using IBM IoT Watson service and node-RED and includes simulated sensors like Temperature, soil moisture, humidity from IBM IoT sensor which can transmit these parameters whenever required.

**Project Requirements:**

Node-RED, IBM Cloud, IBM Watson IoT, Node.js, IBMDevice, IBM IoT Simulator, Python 3.8, Open Weather API platform.

**Project Deliverables:**

Web Application based on IoT based Smart Agriculture System

**1.2 Purpose**

Nowadays monitoring the environmental factor is not the complete solution to increase the yield of crops. There are no. of factors that decrease the productivity to a great extent. Hence Automation must be implemented in agriculture to overcome these problems. An automatic irrigation system thereby saving time, money and power of farmer. The Traditional Farm land irrigation techniques require manual intervention. With the automated technology of irrigation the human intervention can be minimized. Continuous sensing an monitoring of crops by convergence of sensors with Internet of things (IoT) and making farmers to aware about crops growth, harvest time periodically and in turn making high productivity of crops and also ensuring correct delivery of products to end, consumers at right place and right time. So to overcome this problem we go for smart agriculture technique using IOT. It improves the entire agriculture system by monitoring the field in real-time. IoT in agriculture not only saves the time but also reduces the extravagant use of resources such as water and electricity.

**2. LITERATURE SURVEY**

**2.1 Existing Problem**

Agriculture is basic source of livelihood People in India. It plays major role in economy of country. Yet it faces a lot of problems in terms of availability of resources, better irrigation facilities, sudden climatic changes, pests and insects which ruin the crops and thus make a huge loss for this sector.

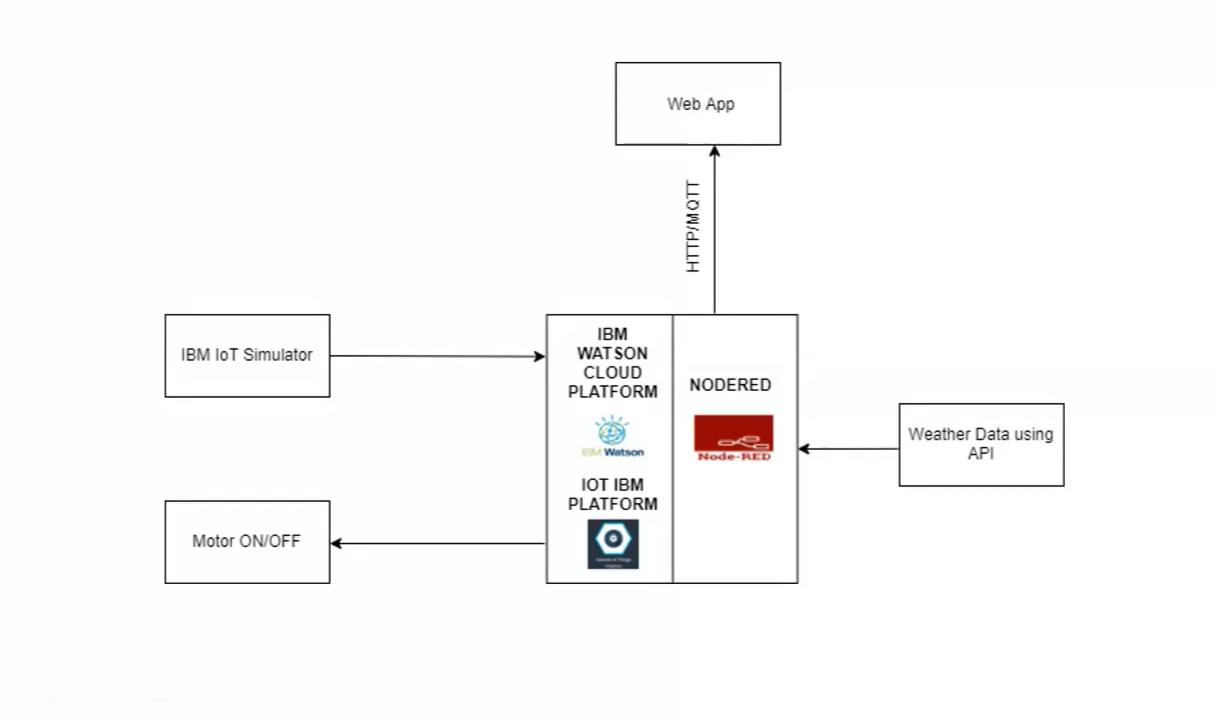
In agriculture it is important to monitor the timely monitoring of irrigating crops. Over supply and under supply, both can have devastating effect on the growth of crops. Also, farmers need to keep track of the soil conditions which is possible only by a physical check. Sometimes crops get damaged due to bad weather which further adds to the worries of the farmers.

**2.2 Proposed Solution**

A Smart Agriculture System based on IoT would be developed which can give real-time data and assist farmers in a very efficient manner. Soil Moisture can be checked by using the sensors that can sense the soil conditions and send the data (moisture content in the soil) over the cloud services to the web application.Based on all the parameters he can water his crop by controlling the motors using the web application. Even if the farmer is not present near his crop he can water his crop by controlling the motors using the web application from anywhere. Surrounding temperature can also be sensed by the sensors and displayed on the web application.

**3. THEORITICAL ANALYSIS**

**3.1 Block Diagram**



**3.2 Hardware/Software Designing**

1. Create a device in IBM Cloud.

2. Connect the device to IBM Simulator to get the weather conditions.

3. Build Node-RED flow to build a web application to display the weather conditions and control the devices.

4. Get the real time weather condition data from open weather map and integrate it in the Node-RED.

5. Control the working of the web application to the devices by python coding.

**4. EXPERIMENTAL INVESTIGATIONS**

1. Completed kick off document.

2. Prepared Project Scope Document with following headings.

Project Summary

Project Requirements

Functional Requirements

Technical Requirements

Software Requirements

Project Deliverables

Project Team

Project Schedule

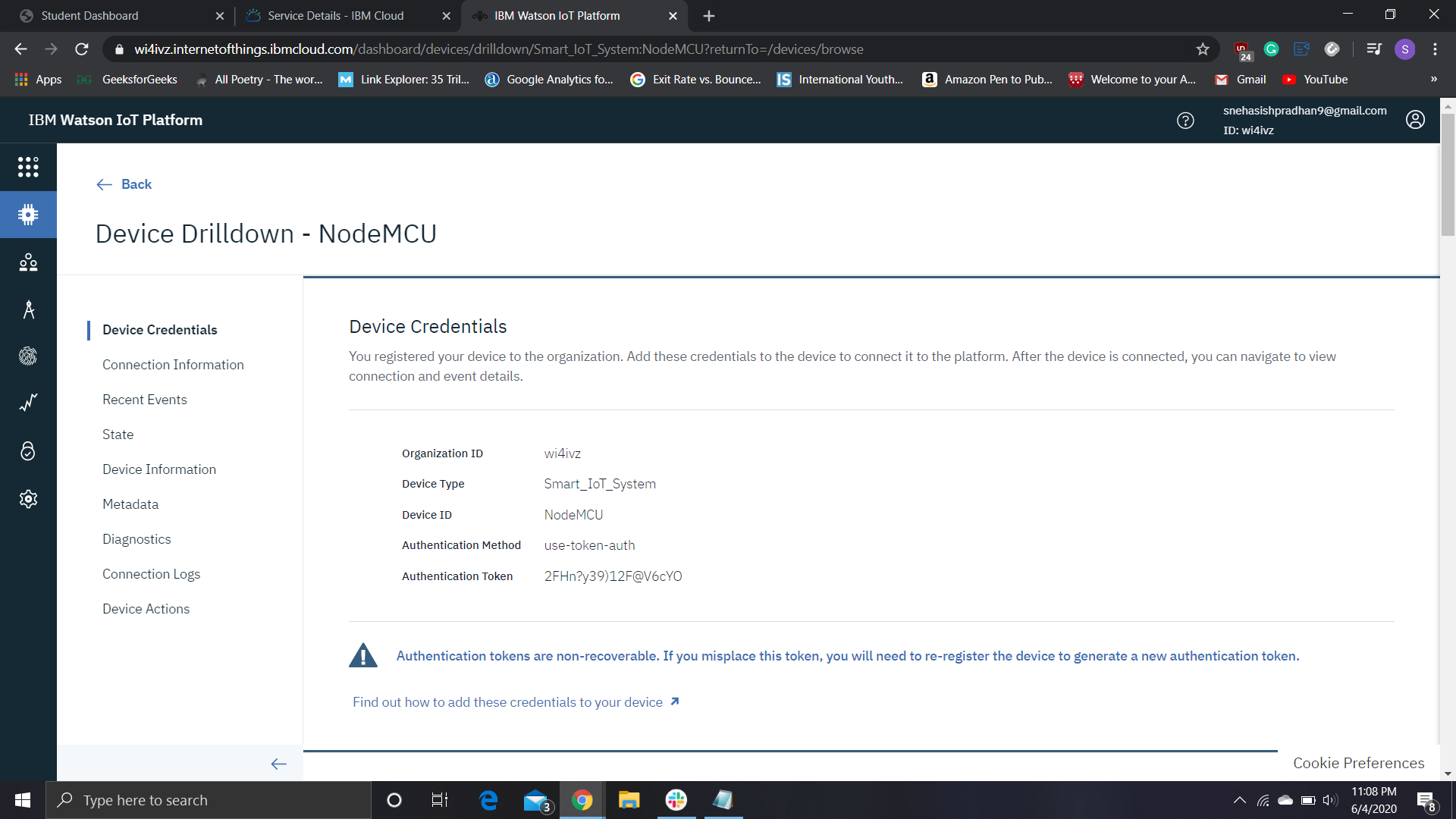
3. Created GitHub account and successfully collaborated with Smart practice school.

4. Created IBM Cloud account and signed up.

5. Installed Node-RED

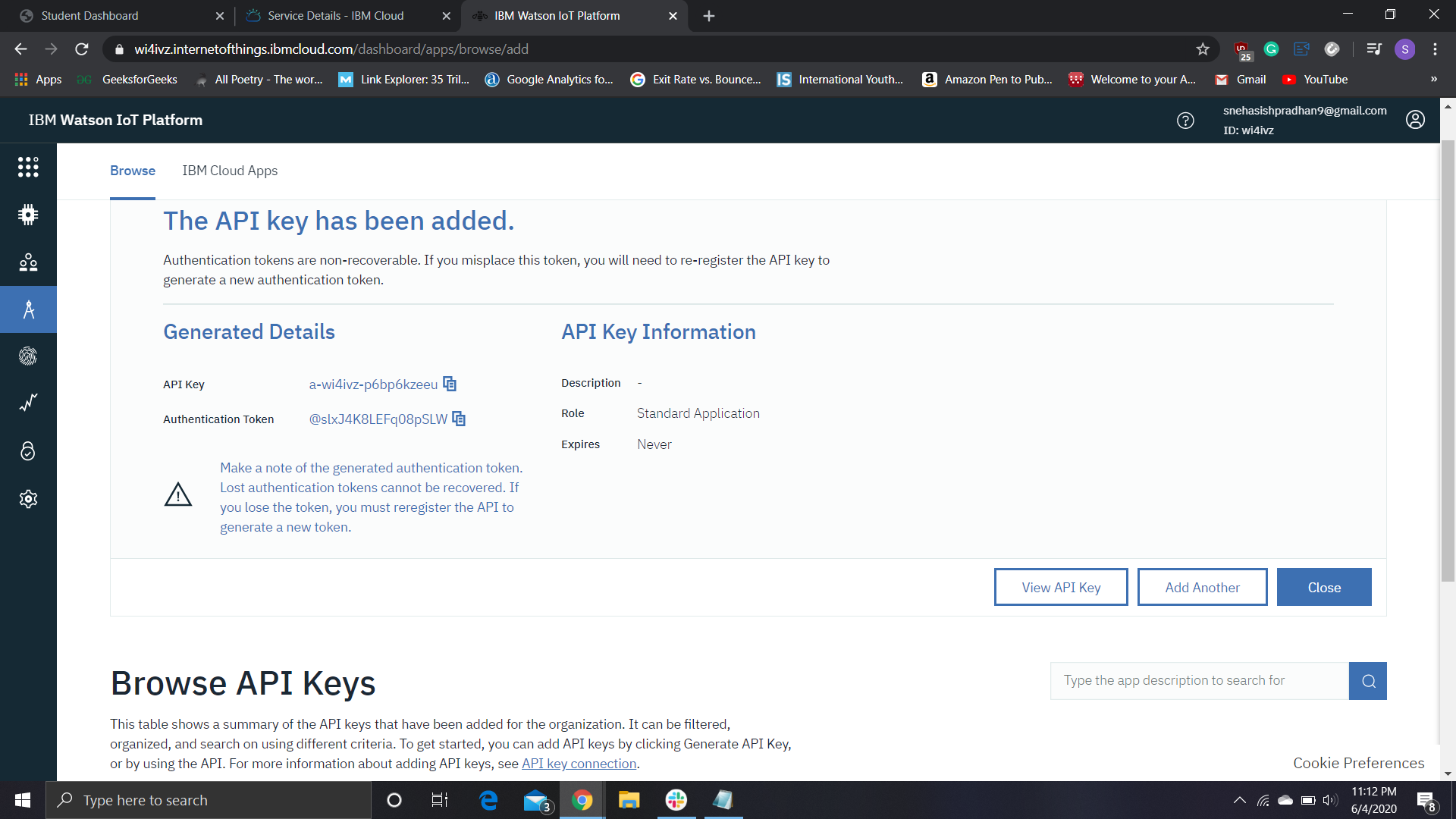
6. Opened Internet of Things service and launched IBM Watson.

7. Created a device in IBM Watson



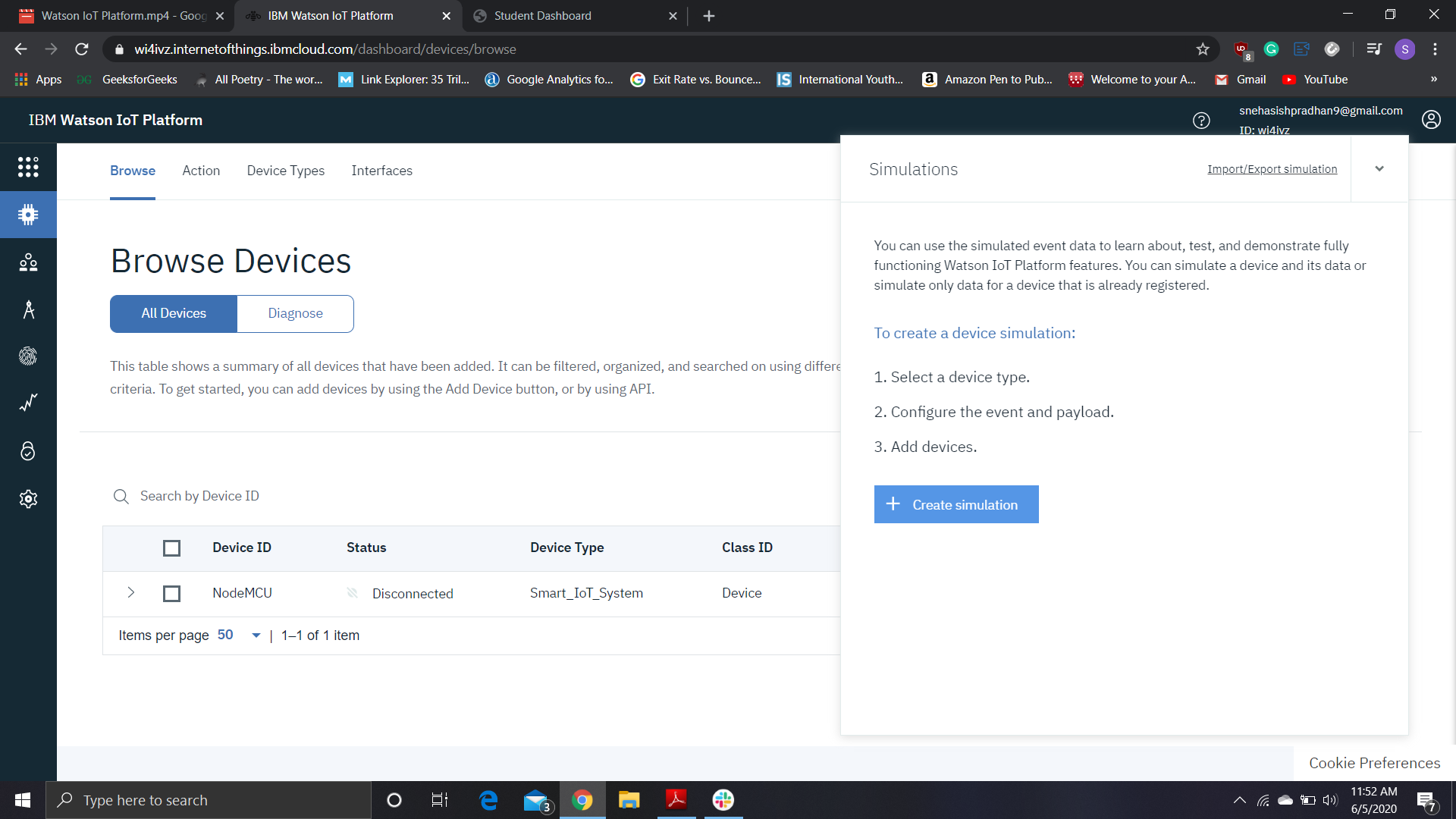
8. Installed Python 3 Idle.

9. Generated API Key in IBM Watson IOT Platform.

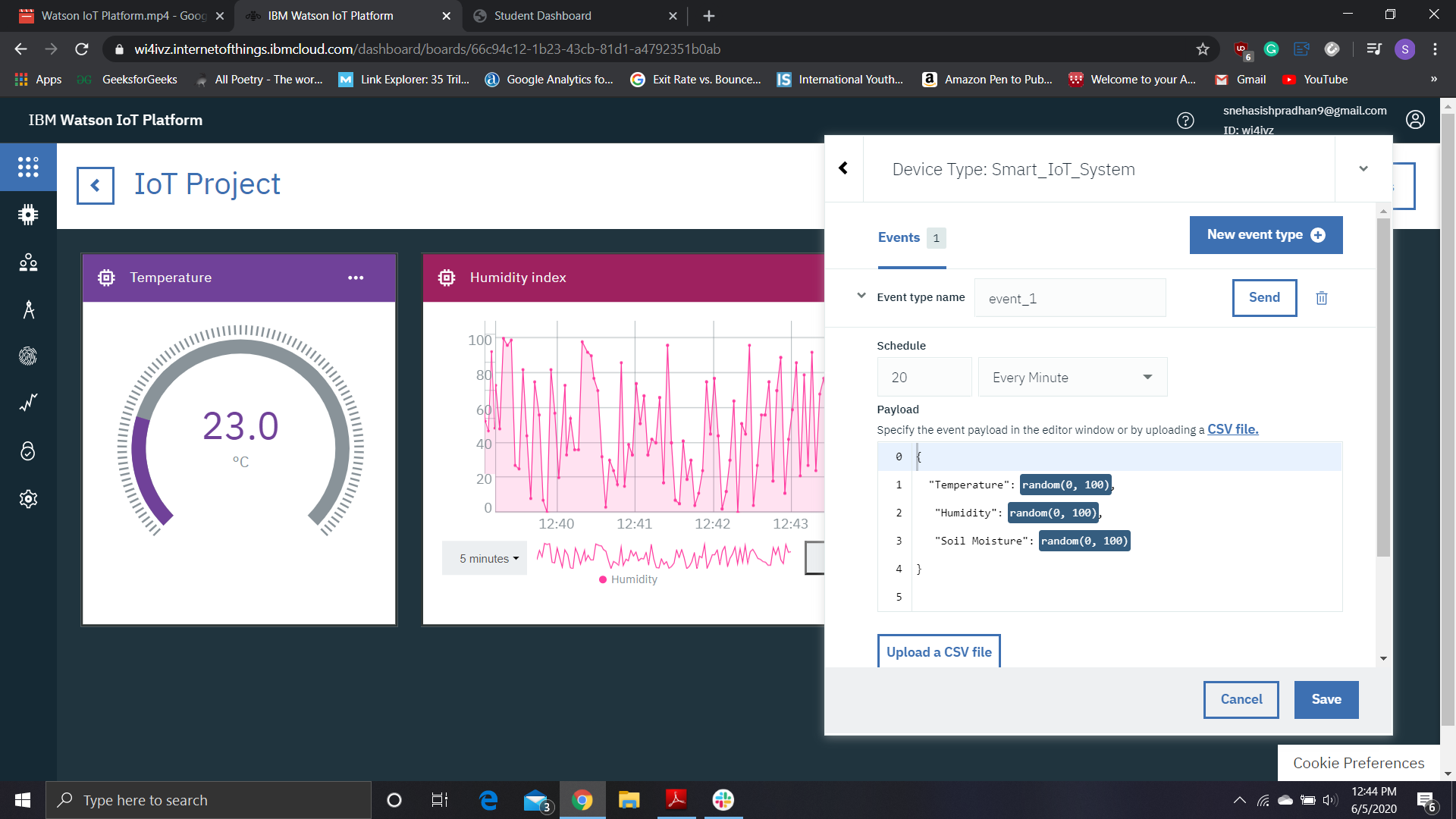


-> Keep all the details in the notepad for future use.

10. Simulated the device created by specifying event payload.

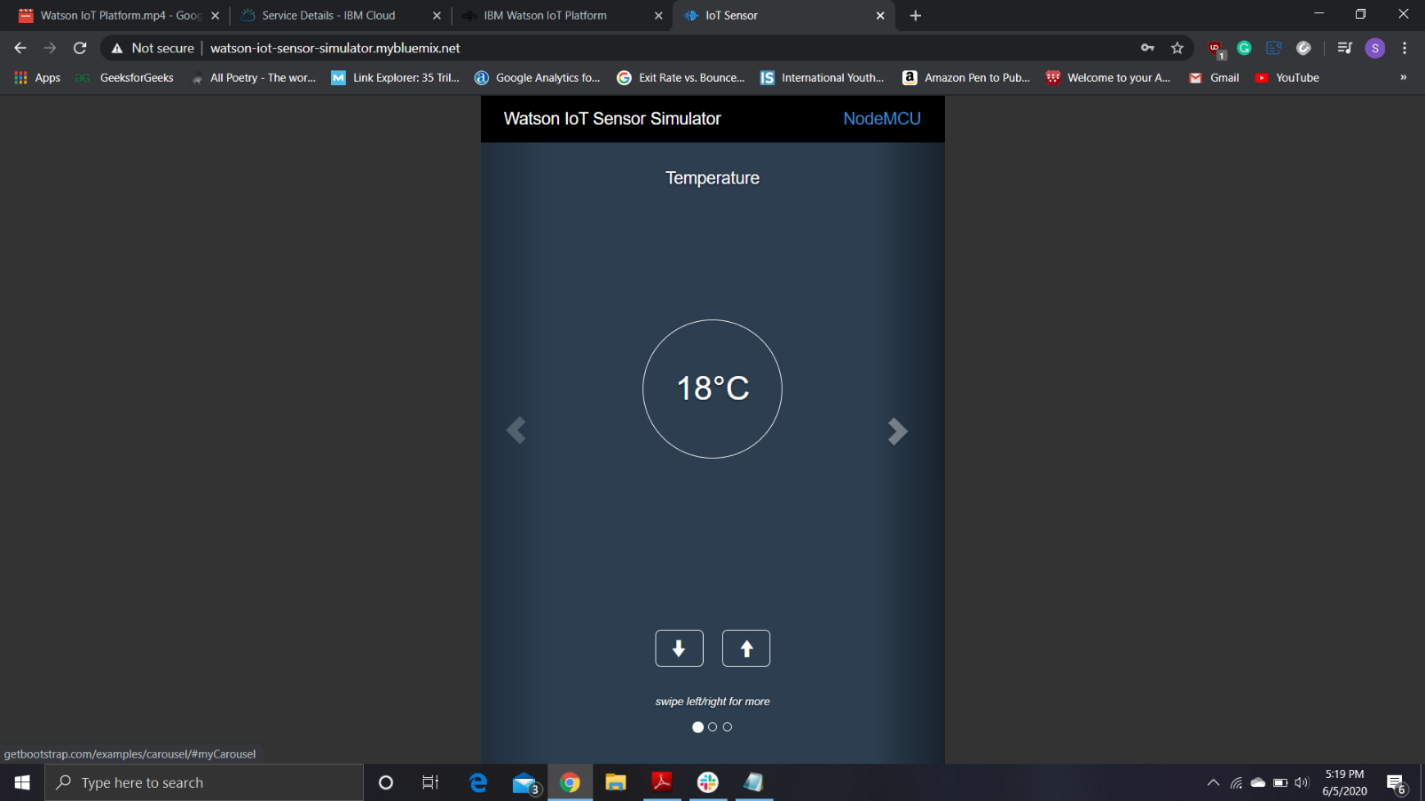


11. Created dashboards for visualizing charts for event\_1 simulation.

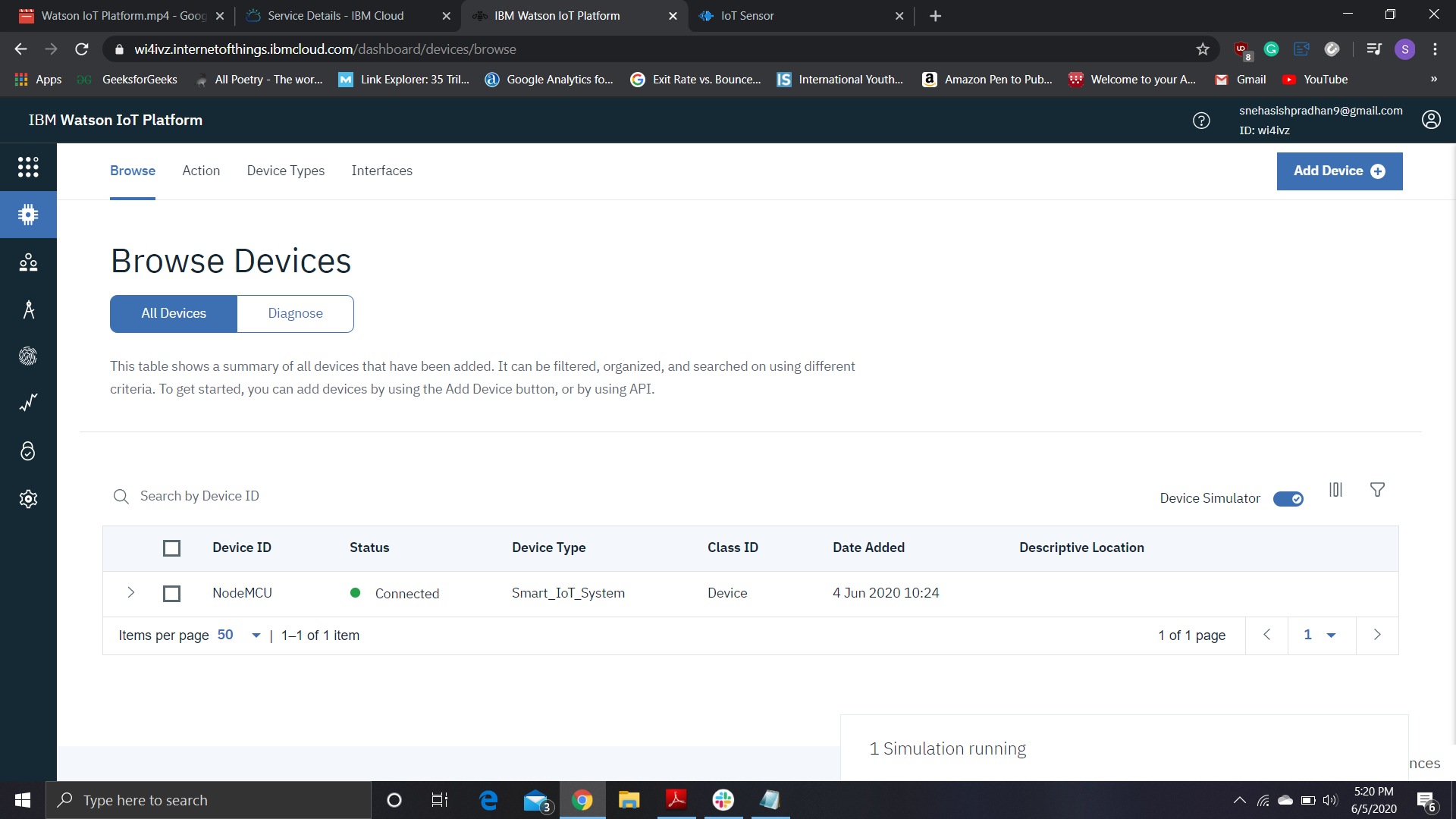


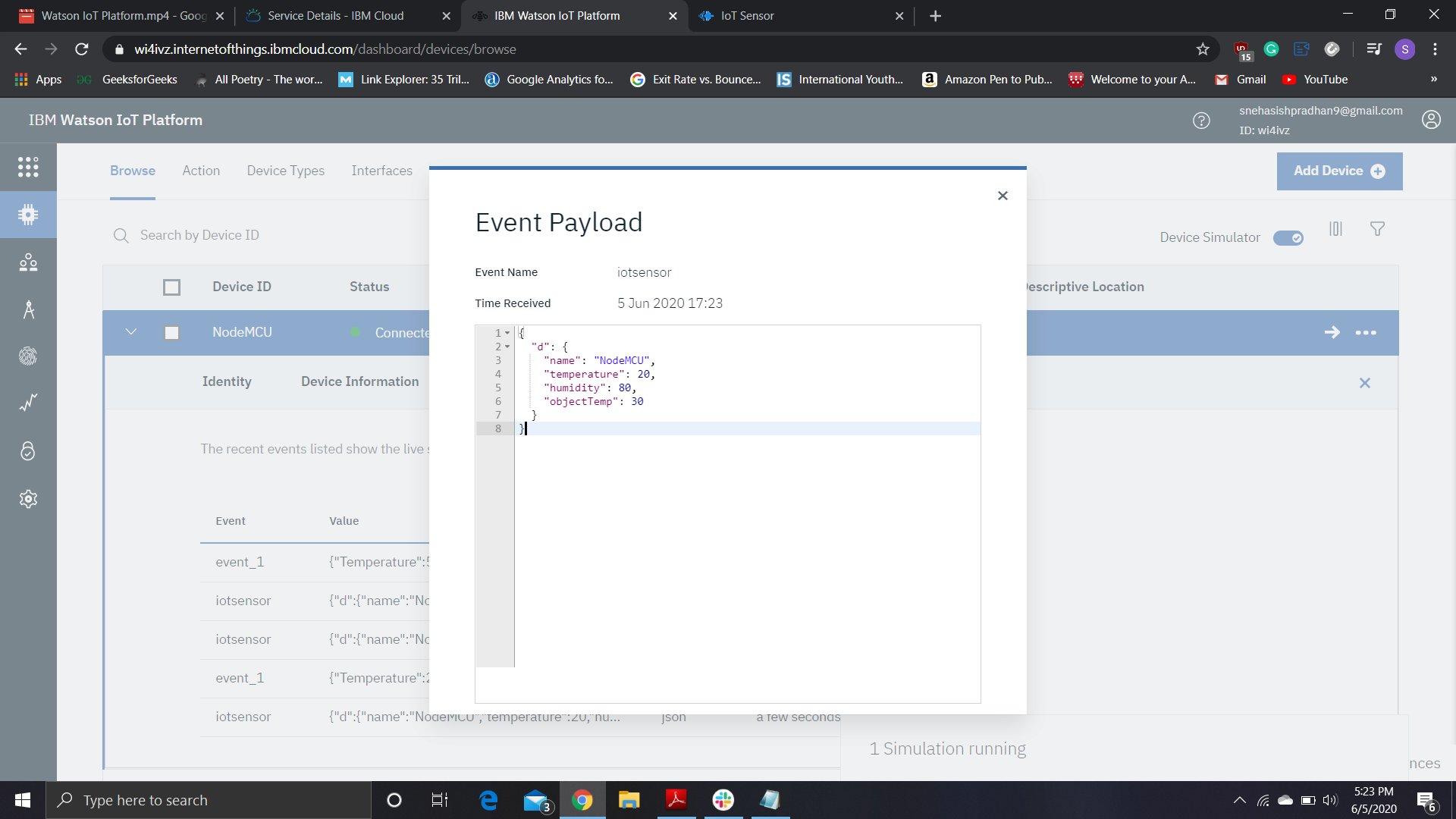
12. Opened IBM IoT sensor simulator and filled relevant credentials.

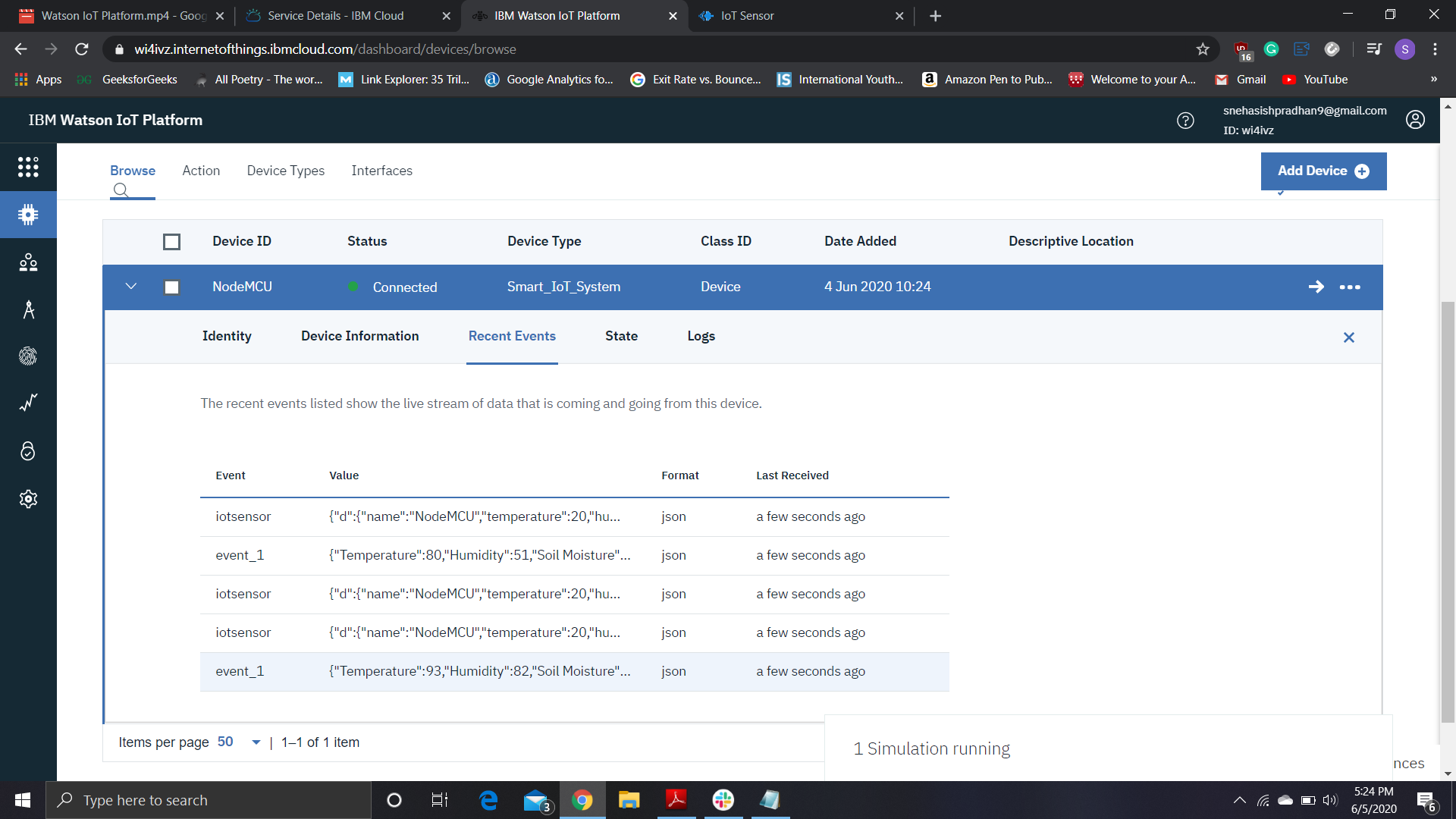
Successfully connected….



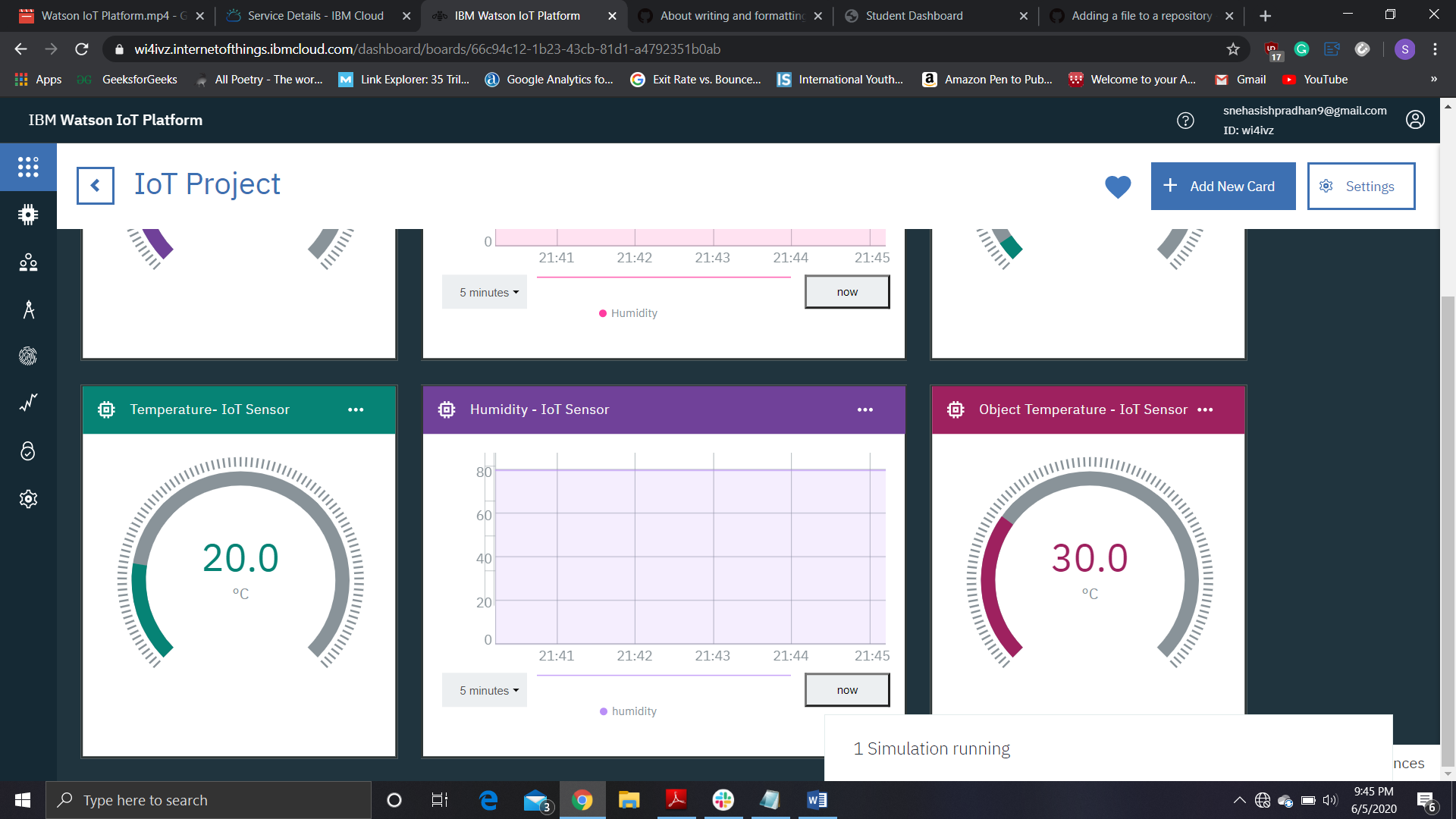
13. The status of the device is now showing connected and data is appearing under recent events.



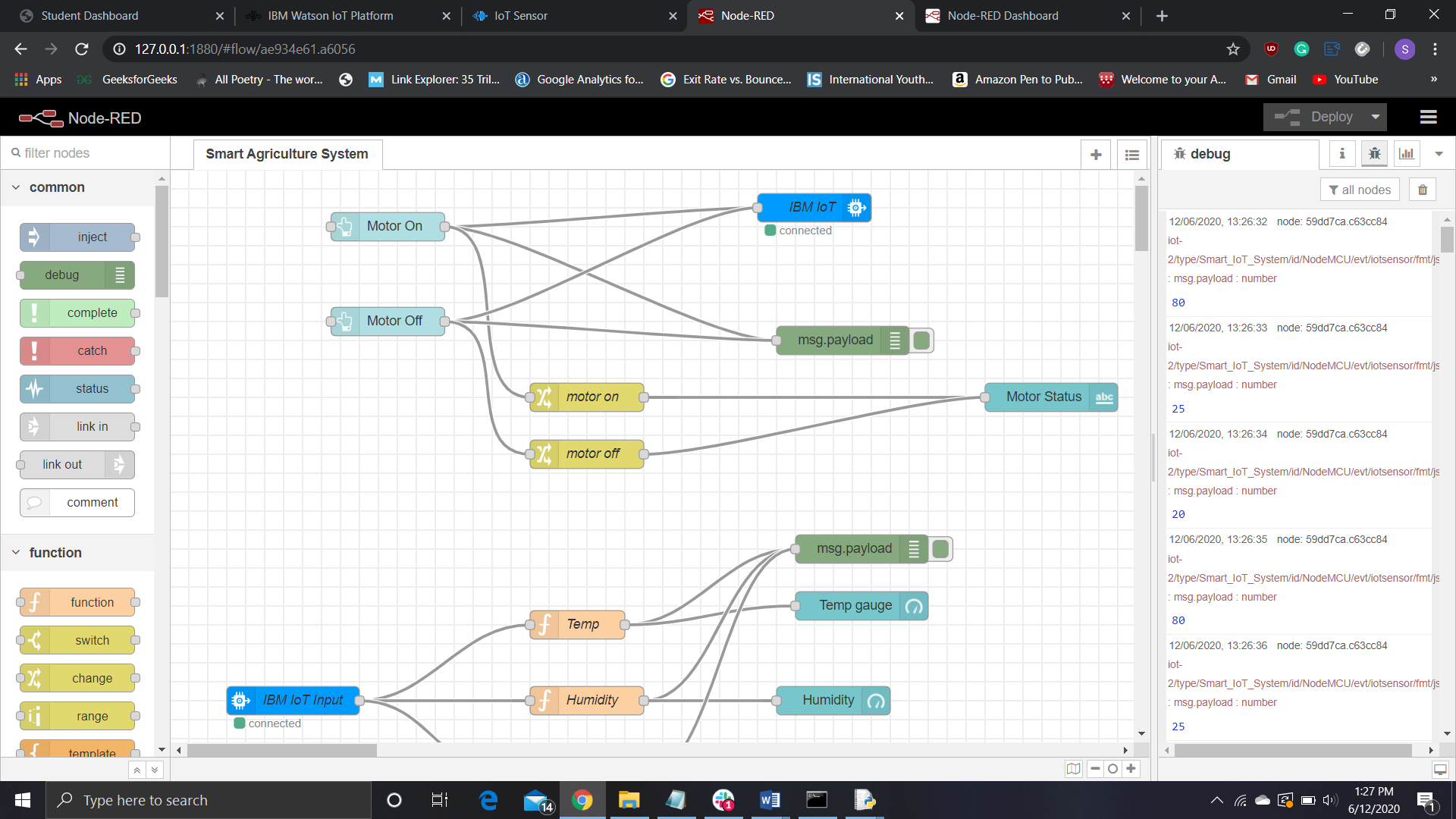


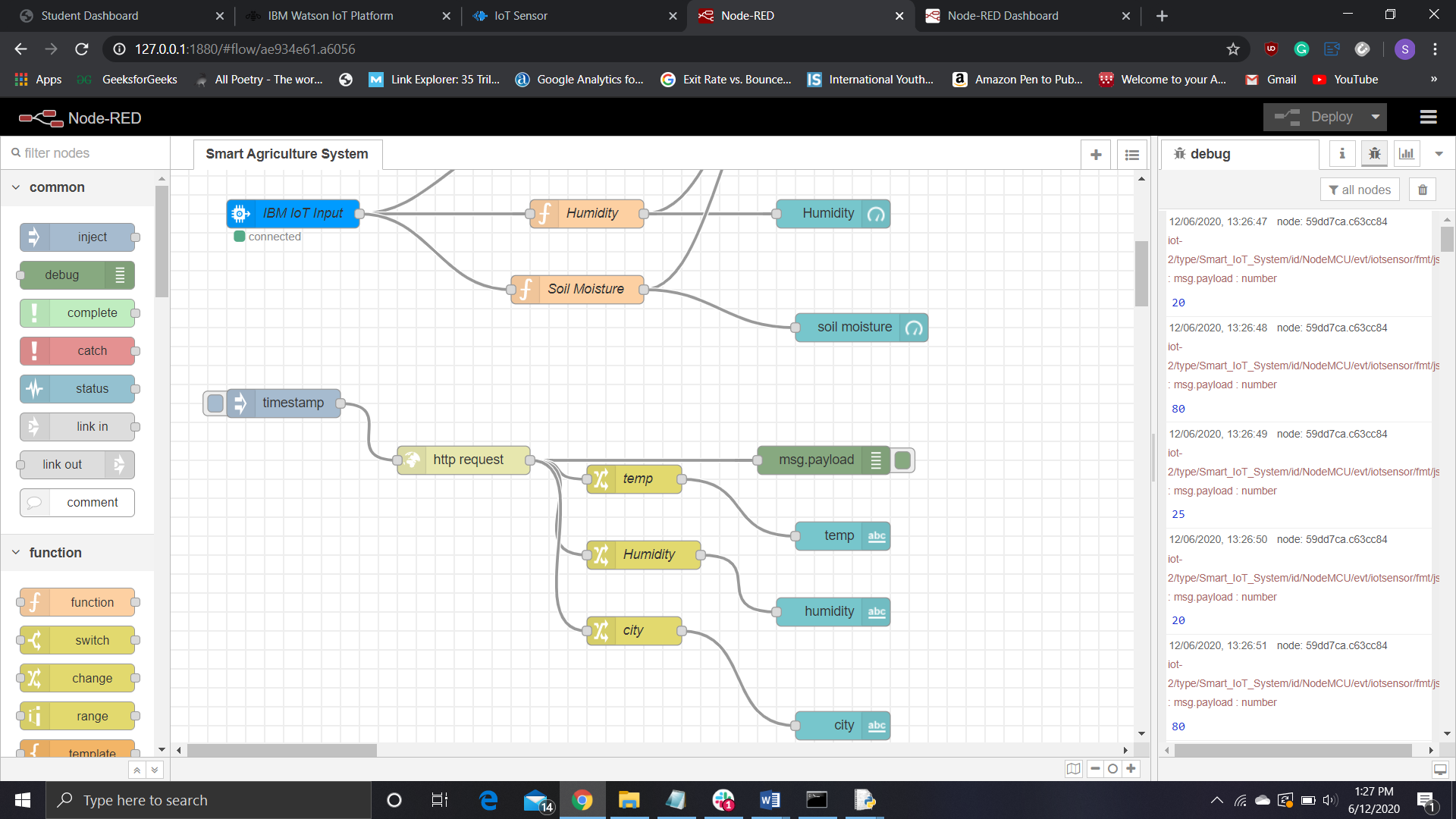


14. Created some more cards in the dashboard for IoT sensor simulator data.

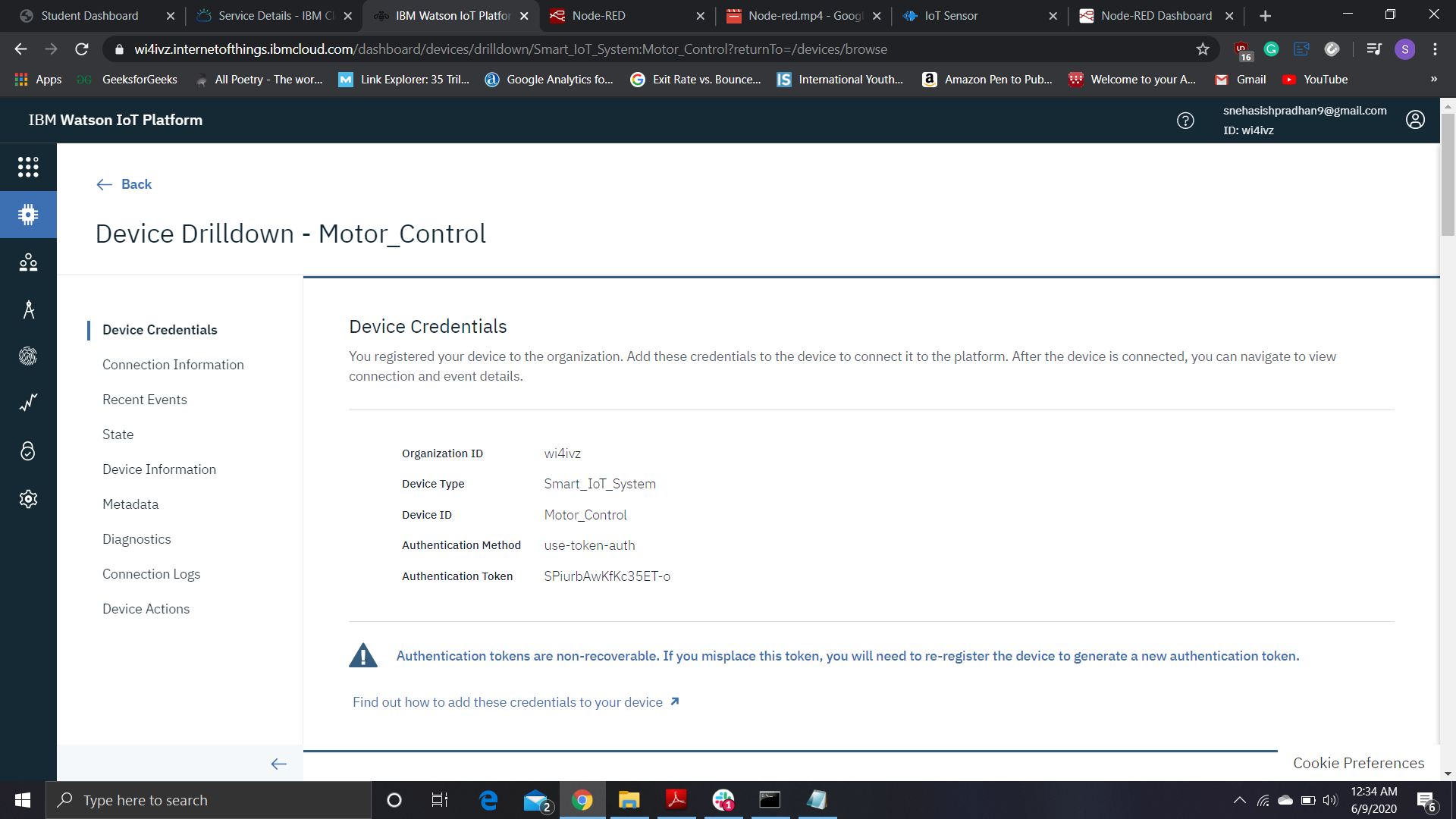


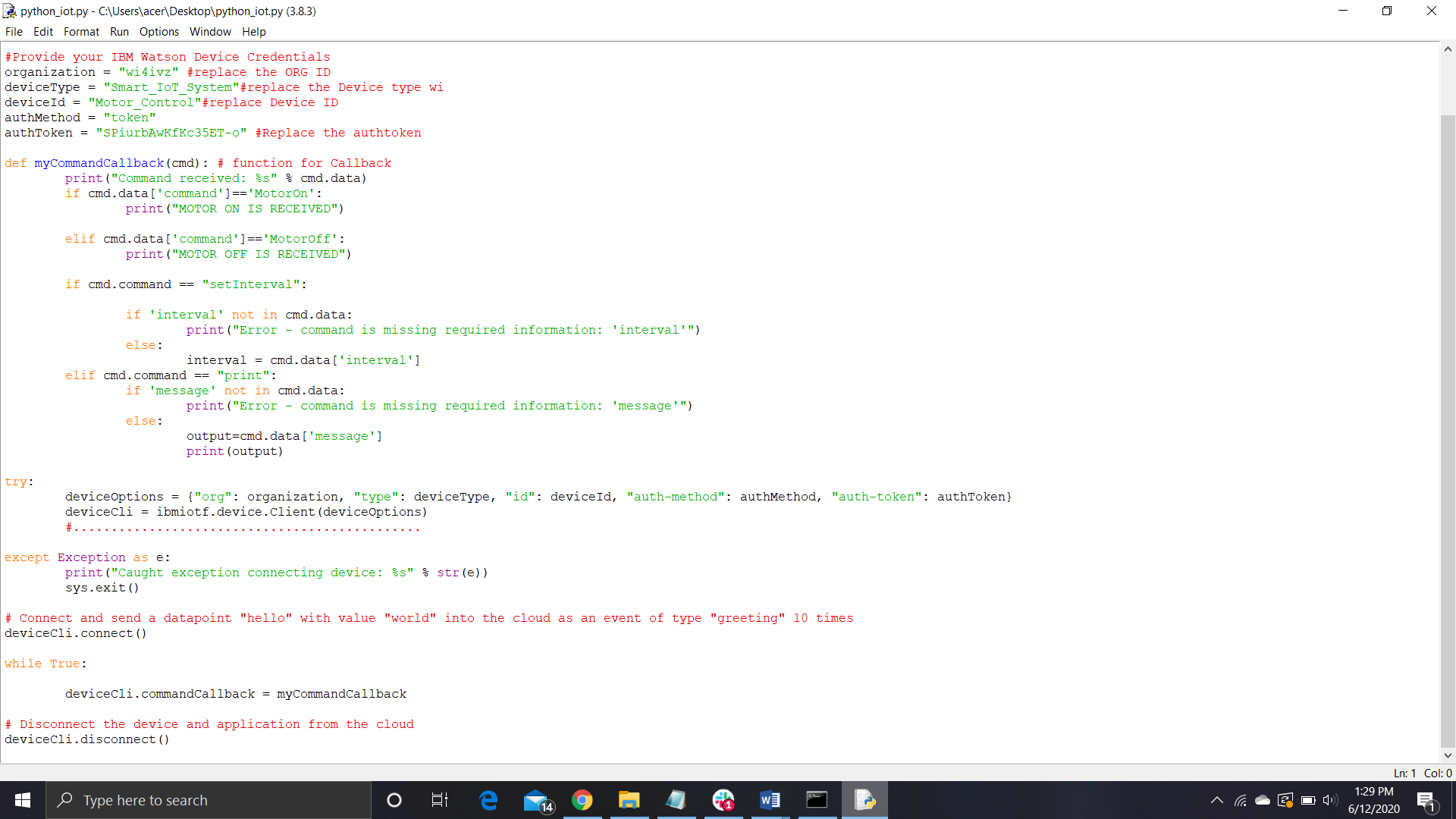
15. Opened node-RED and created the flow using different nodes. All the nodes are appropriately connected and provided with correct credentials.



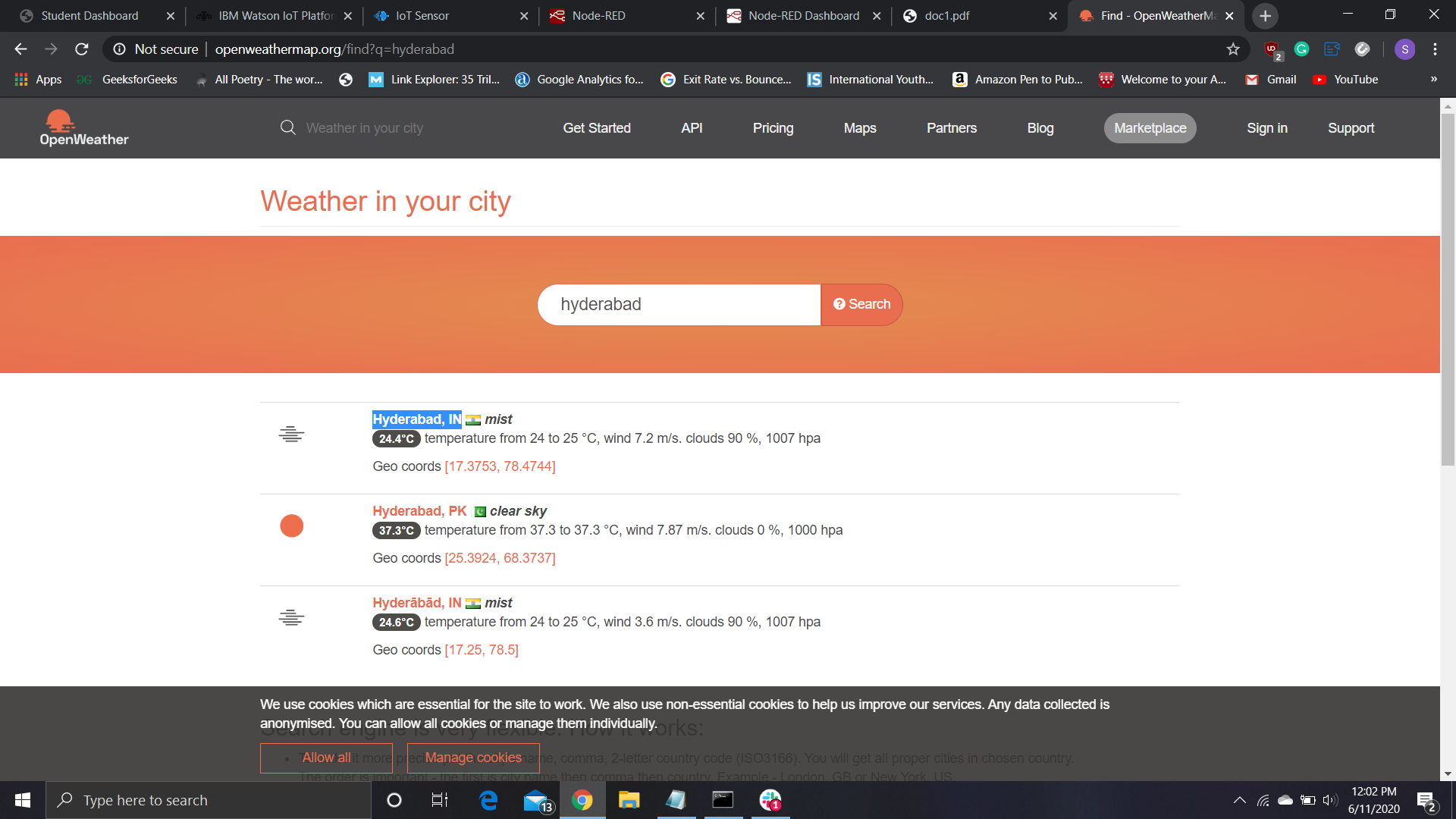


16. For Motor on/off provision another device is created and connected to node-red. A python program is written in order to receive motor command details.

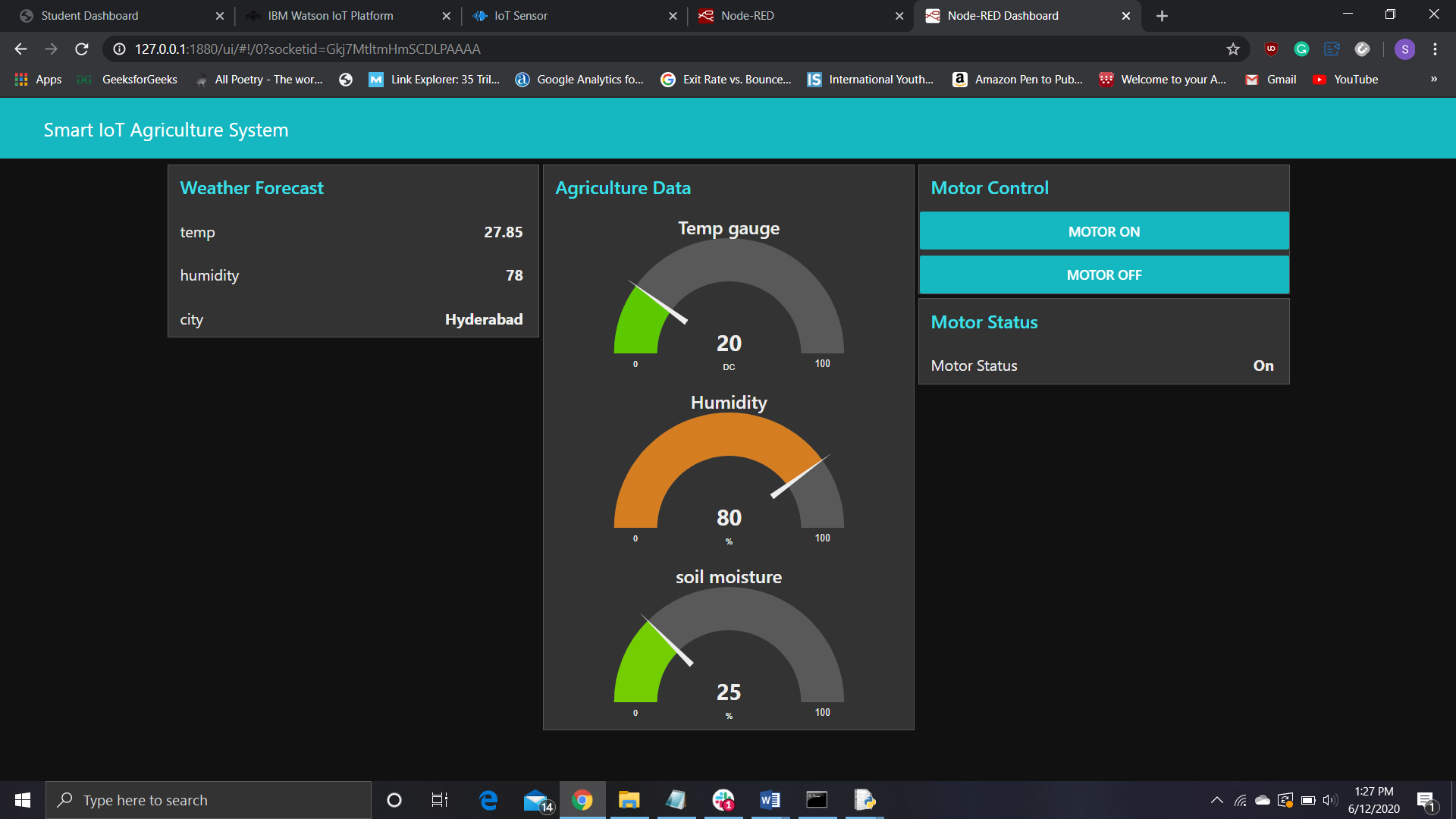


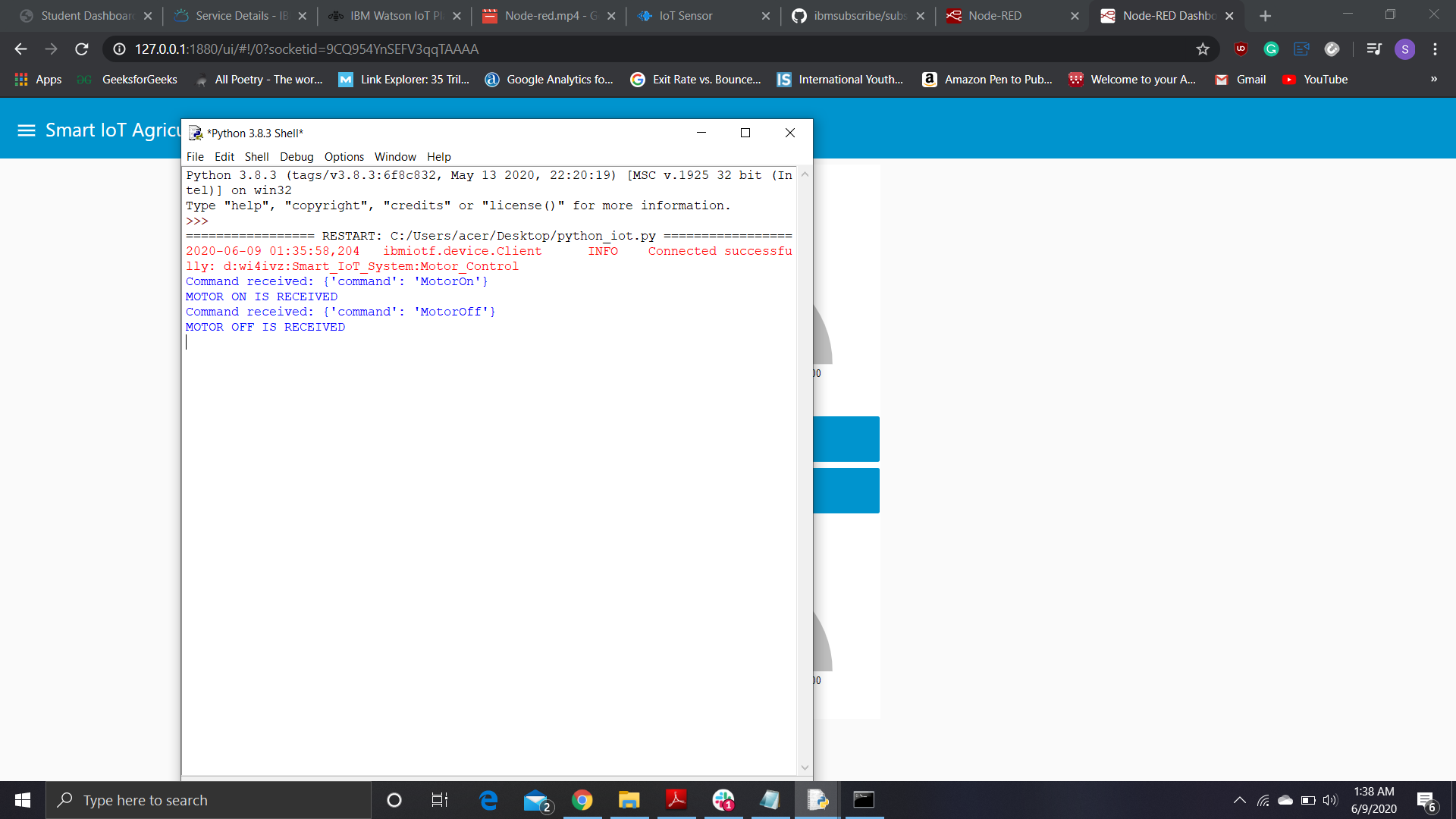


17. To show the current weather forecast of the location, Open Weather API is used. The process involves creating an account, finding the city name, getting the API key and finally the URL for http request.



18. Finally the model is deployed and web app dashboard is displayed. Here are some snapshots of its working.

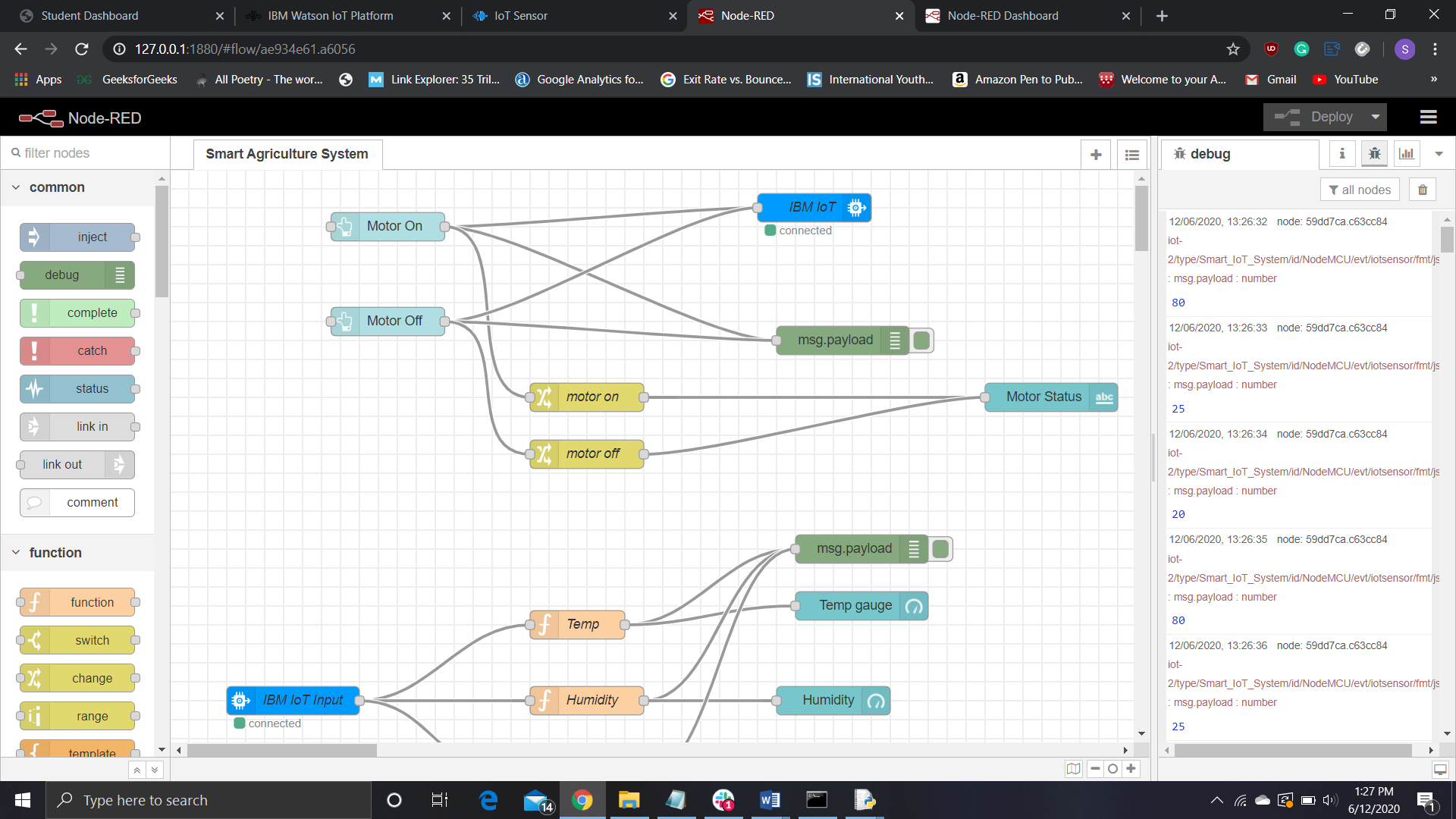


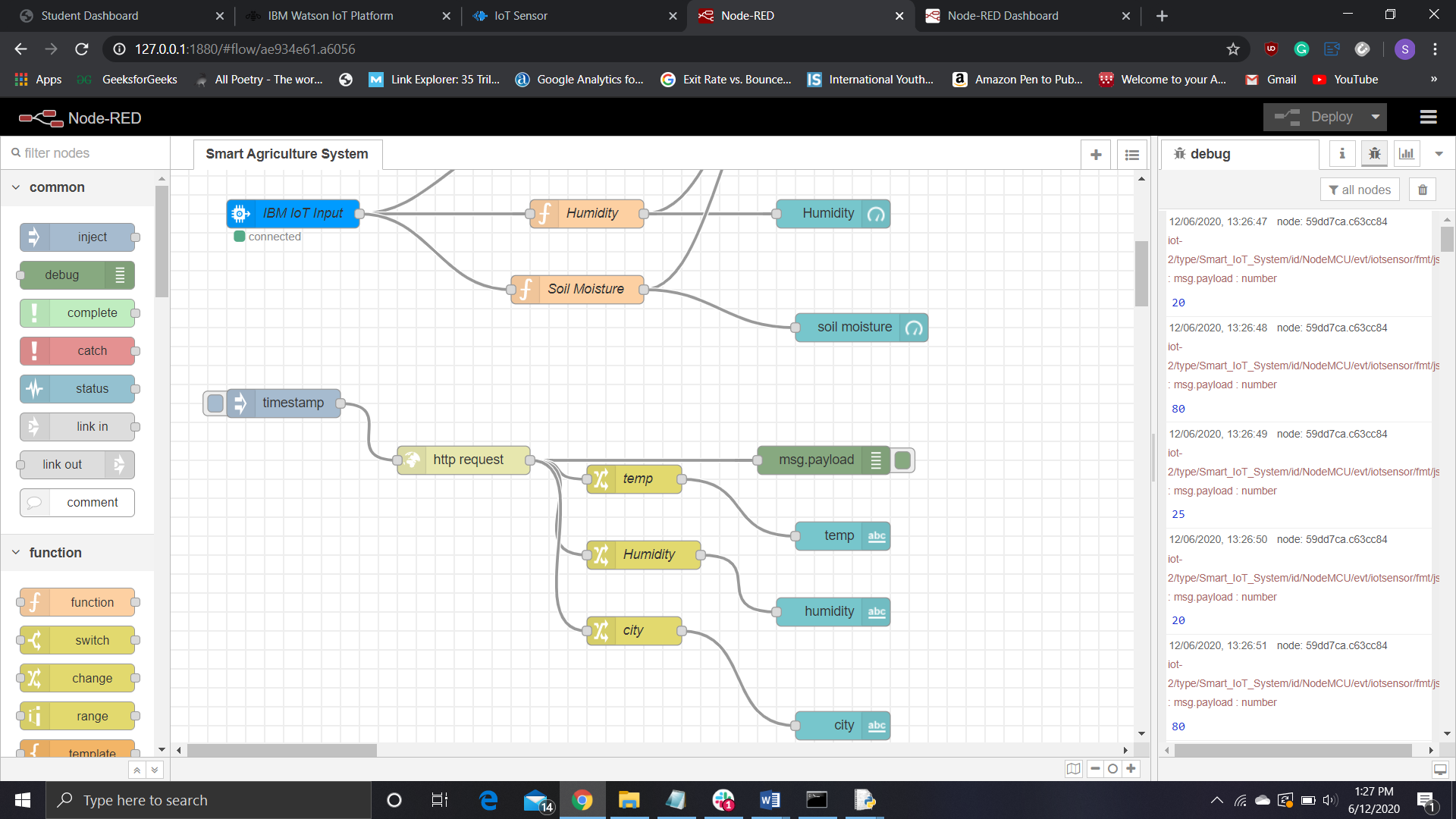


**5. FLOWCHART**

In this project we the agriculture related data through IBM IoT Simulator instead of real soil and temperature conditions. Simulator passes the data through IBM Cloud to the web application. The data is displayed on the dashboard and Web Application is build using Node-RED. We

Web Application is also used to control the devices further like motor, pumps, lights, or any other devices in the agricultural field. In this project the output is passed using python code and the control action is displayed in python code console window.

****

****

**6. RESULT**

We have successfully built an interactive web-based UI and integrated all the services using Node-RED and IBM IoT Watson. The dashboard displays the current weather forecast of the location, agriculture based data like temperature, humidity, soil moisture and motor on/off controls.

**7. ADVANTAGES AND DISADVANTAGES**

**7.1 Advantages**

1. All the data like crop and soil conditions including current weather information can easily be viewed anywhere and anytime through this mobile/web app.

2. Monitoring crop and soil through this way can reduce crop damage to a greater extent.

3. Laborious and time taking tasks can be done with a single touch in just a second.

4. Accepting and implementing this technique can lead to increase in overall yield and quality of products.

**7.2 Disadvantages**

1. Smart Agriculture System requires continuous internet connectivity which is very difficult to maintain in rural areas.

2. Most of the rural people lack smartphones and any such web devices.

3. Any fault in IoT sensors can lead to malfunctioning of agricultural activities and whole automated process would be highly affected.

4. Regular maintenance of the devices are required and only IoT specialized people can solve the problem.

**8. APPLICATION**

Introduction of IoT is a boon which has the ability to transform the face of this sector completely in the upcoming years. Farming and other agricultural activities can be carried out in precise and accurate manner. The quality and quantity of crops produced would be high. Therefore the whole agriculture sector would be highly benefitted.

**9. CONCLUSION**

An IoT Web Application is built successfully for smart agricultural system using Watson IoT platform, Watson simulator, IBM cloud and Node-RED. This will assist farmers in getting live data (Temperature, soil moisture, humidity) for efficient environment monitoring which will enable them to do smart farming and increase their overall yield and quality of products.

This app intends to provide farmers with a mobile app using which they can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details. Based on all the parameters he can water his crop by controlling the motors using the web application. Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application from anywhere.

**10. FUTURE SCOPE**

With the growing population of the nation, the demand for food also increases and to satisfy all the needs, it is important that agriculture activities are carried out in most accurate and efficient manner. Bringing in IOT technology over here can assist us greatly with automated farming activities, maintaining the quality of the products and reducing the extravagant use resources like electricity and water.

**11. BIBILOGRAPHY**

1. <https://www.ibm.com/in-en/cloud>

2. <https://www.ibm.com/in-en/internet-of-things>

3. <https://flows.nodered.org/node/node-red-dashboard>

[4.https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%20(1).pdf](https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%20(1).pdf)

5.<https://smartinternz.com/assets/docs/Sending%20Http%20request%20to%20Open%20weather%20map%20website%20to%20get%20the%20weather%20forecast.pdf>

6. <https://openweathermap.org/>

7. <https://www.youtube.com/watch?v=cicTw4SEdxk>

8.[https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%2 0(1).pdf](https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Services%252%20%20%20%20%20%200(1).pdf)

**12. APPENDIX**

**A) Source Code**

Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | import time |
|  | import sys |
|  | import ibmiotf.application # to install pip install ibmiotf |
|  | import ibmiotf.device |
|  |  |
|  | #Provide your IBM Watson Device Credentials |
|  | organization = "Org ID" #replace the ORG ID |
|  | deviceType = "Device Type"#replace the Device type wi |
|  | deviceId = "Device ID"#replace Device ID |
|  | authMethod = "token" |
|  | authToken = "authtoken" #Replace the authtoken |
|  |  |
|  | def myCommandCallback(cmd): # function for Callback |
|  | print("Command received: %s" % cmd.data) |
|  | if cmd.data['command']=='lighton': |
|  | print("LIGHT ON IS RECEIVED") |
|  |  |
|  | elif cmd.data['command']=='lightoff': |
|  | print("LIGHT OFF IS RECEIVED") |
|  |  |
|  | if cmd.command == "setInterval": |
|  |  |
|  | if 'interval' not in cmd.data: |
|  | print("Error - command is missing required information: 'interval'") |
|  | else: |
|  | interval = cmd.data['interval'] |
|  | elif cmd.command == "print": |
|  | if 'message' not in cmd.data: |
|  | print("Error - command is missing required information: 'message'") |
|  | else: |
|  | output=cmd.data['message'] |
|  | print(output) |
|  |  |
|  | try: |
|  | deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken} |
|  | deviceCli = ibmiotf.device.Client(deviceOptions) |
|  | #.............................................. |
|  |  |
|  | except Exception as e: |
|  | print("Caught exception connecting device: %s" % str(e)) |
|  | sys.exit() |
|  |  |
|  | # Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times |
|  | deviceCli.connect() |
|  |  |
|  | while True: |
|  |  |
|  | deviceCli.commandCallback = myCommandCallback |
|  |  |
|  | # Disconnect the device and application from the cloud |
|  | deviceCli.disconnect() |