

SMARTBRIDGE: PROJECT

Predictive Maintenance Of Industrial Motors Using IBM Cloud

TEAM MEMBERS:

1. P. Mohan Sai

mohansai.pragada2019@vitstudent.ac.in

2. S. Abishek

abishek.s2019@vitstudent.ac.in

3. P.V.N.S Sai Saran

Paramathmuni.saisaran2019@vitstudent.ac.in

INTRODUCTION:

OVERVIEW:

To predict the industrial motor will work or won't, we can achieve this by measuring the values of the motors.

OBJECTIVE:

We have to measure the values of current, voltage, temperature. We can measure this characters by using ML, a machine learning model can be developed by using IBM CLOUD. To develop it we are using AUTO AI mode of IBM cloud.

SURVEY:

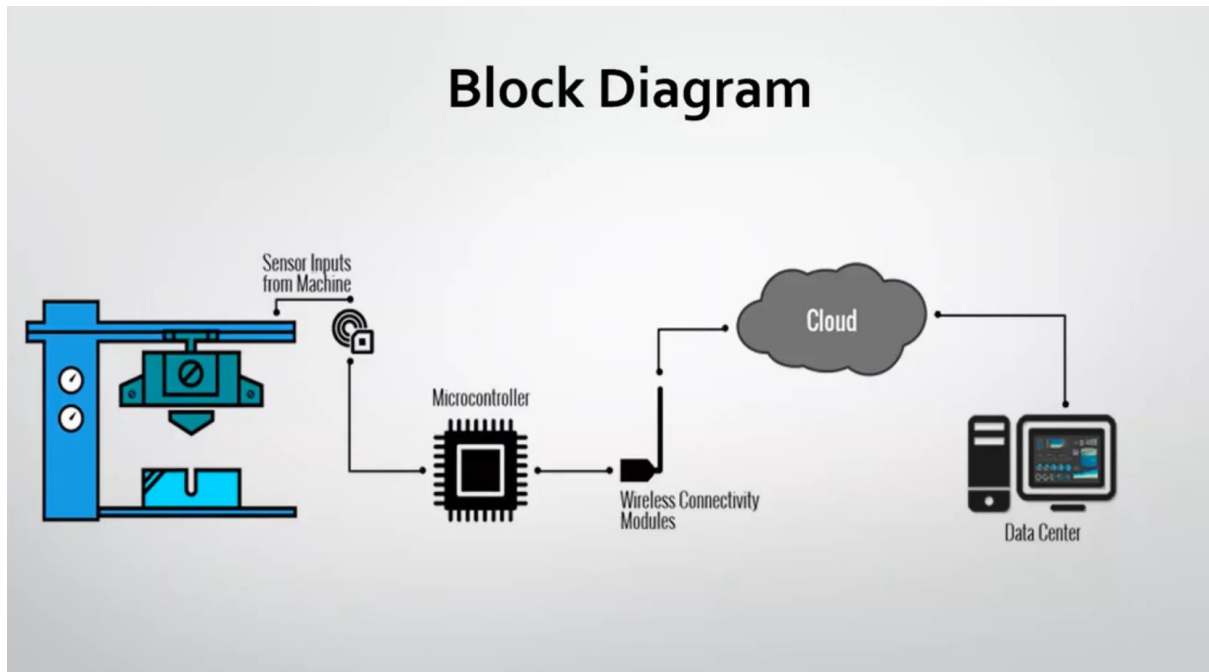
a. Existing model:

General model which can detect without proper data.

b. **Solution:**

We can integrate this model with IBM cloud with introducing to AUTO AI .to get the data correctly

BLOCK DIAGRAM:



REQUIREMENTS:

Python IDLE

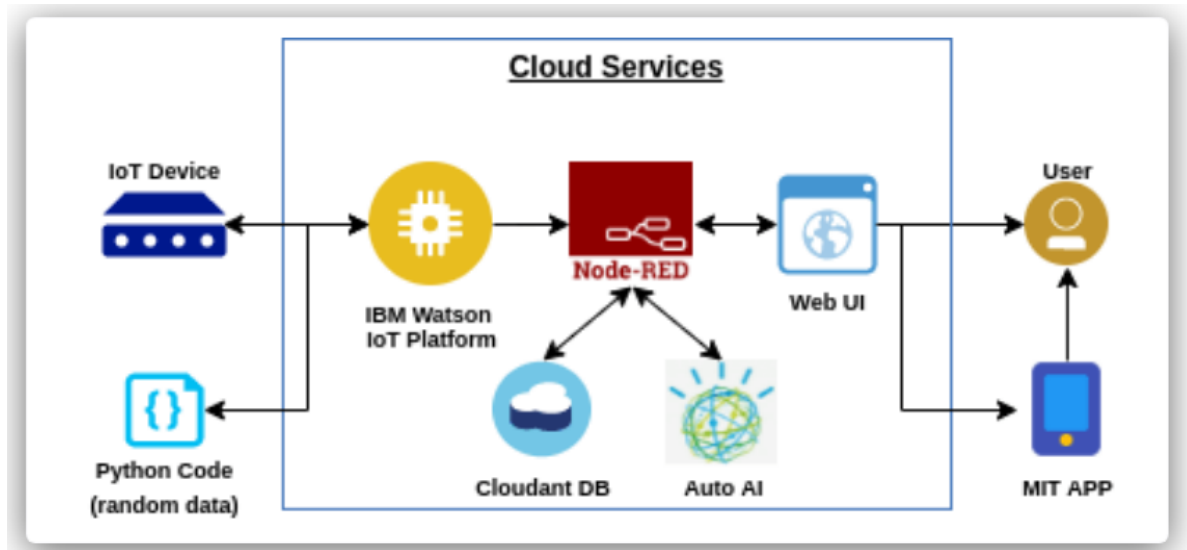
IBM ACCOUNT

Node-red

AUTO AI

MIT APP

FLOW CHART:



PROCEDURE:

Develop the code

```
iotbm.py - C:\Users\DELL\AppData\Local\Programs\Python\Python39\iotbm.py (3.9.6)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "kj14w7",
        "typeId": "VITElectrical",
        "deviceId": "69510"
    },
    "auth": {
        "token": "128951045"
    }
}

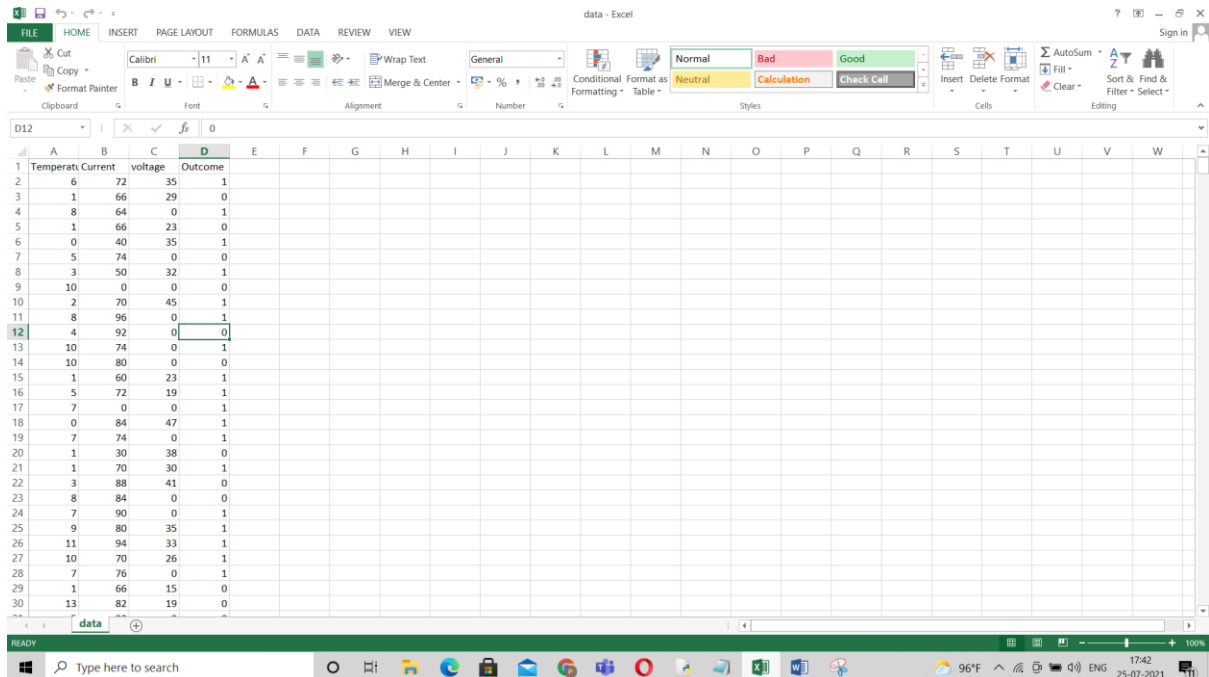
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    print(m)
    if m == "MOTOR ON":
        print("MOTOR is on")
    elif m == "MOTOR OFF":
        print("MOTOR is off")
    print()

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(0,110)
    curr=random.randint(0,110)
    voll=random.randint(0,110)
    myData={'temperature':temp, 'current':curr, 'voltage':voll}
    client.publishEvent(eventId="DHT11", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

Creating the AUTO-AI application

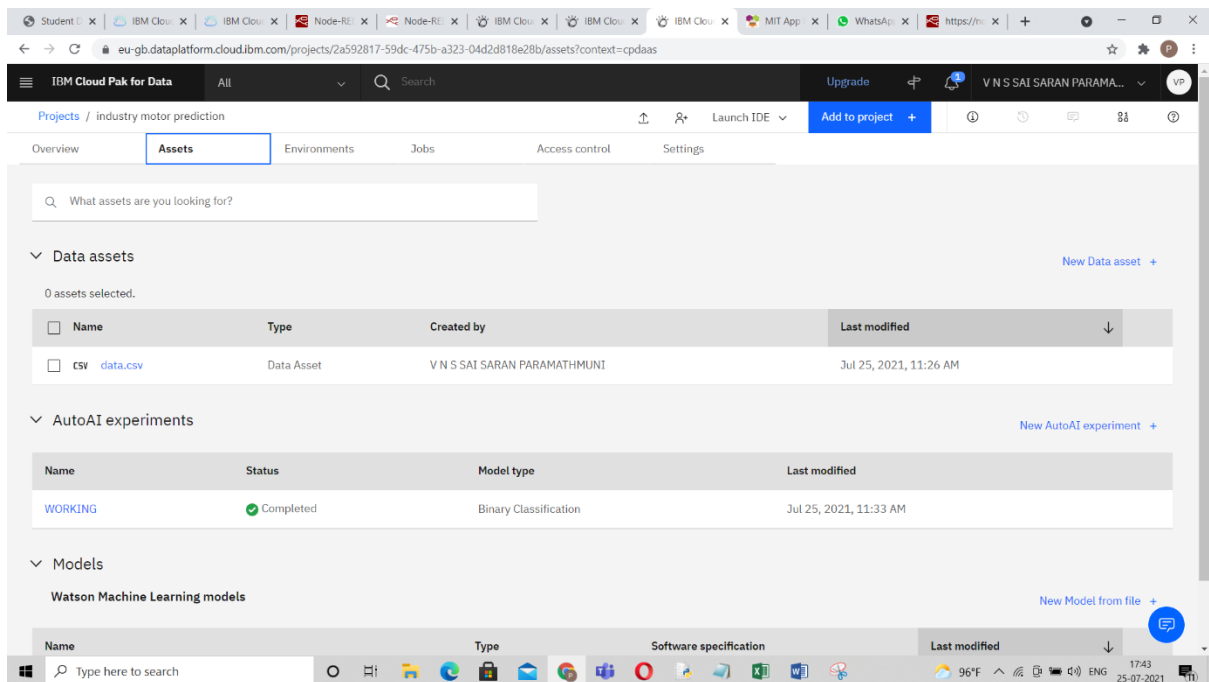
To get the application we need to add a dataset of the motor



The screenshot shows a Microsoft Excel spreadsheet with a dataset of motor performance data. The data is organized into columns: A (Temperature), B (Current), C (voltage), and D (Outcome). The rows represent individual data points, with the first row (row 1) serving as the header. The data is as follows:

Temperature	Current	voltage	Outcome
6	72	35	1
1	66	29	0
8	64	0	1
1	66	23	0
0	40	35	1
5	74	0	0
3	50	32	1
10	0	0	0
2	70	45	1
8	96	0	1
4	92	0	0
10	74	0	1
10	80	0	0
1	60	23	1
5	72	19	1
7	0	0	1
0	84	47	1
7	74	0	1
1	30	38	0
1	70	30	1
3	88	41	0
8	84	0	0
7	90	0	1
9	80	35	1
11	94	33	1
10	70	26	1
7	76	0	1
1	66	15	0
13	82	19	0

After giving the data set we will get our ML page in IBM Cloud



The screenshot shows the IBM Cloud Pak for Data interface. The 'Assets' tab is selected, displaying a list of data assets. The first asset is a CSV file named 'data.csv', which is a Data Asset created by V N S SAI SARAN PARAMATHMUNI on Jul 25, 2021, at 11:26 AM. Below the data assets, there is a section for 'AutoAI experiments' showing a single experiment named 'WORKING' with a status of 'Completed' and a model type of 'Binary Classification', last modified on Jul 25, 2021, at 11:33 AM. The interface also includes a search bar and various navigation options.

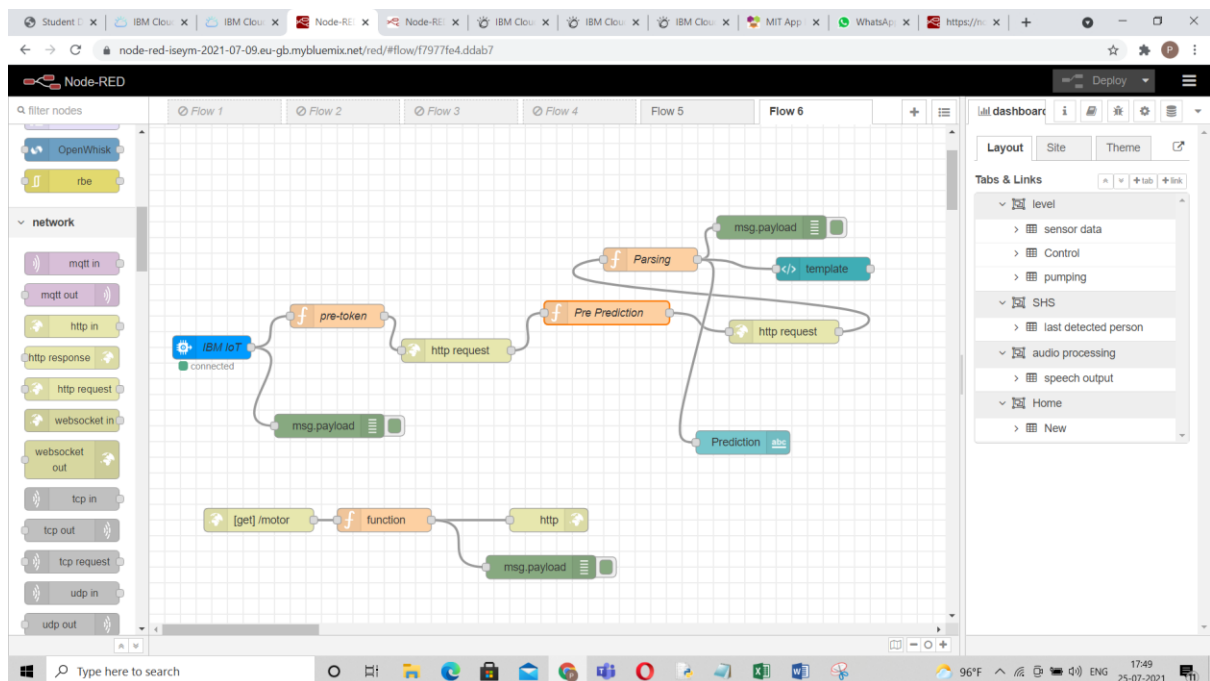
Name	Type	Created by	Last modified
data.csv	Data Asset	V N S SAI SARAN PARAMATHMUNI	Jul 25, 2021, 11:26 AM

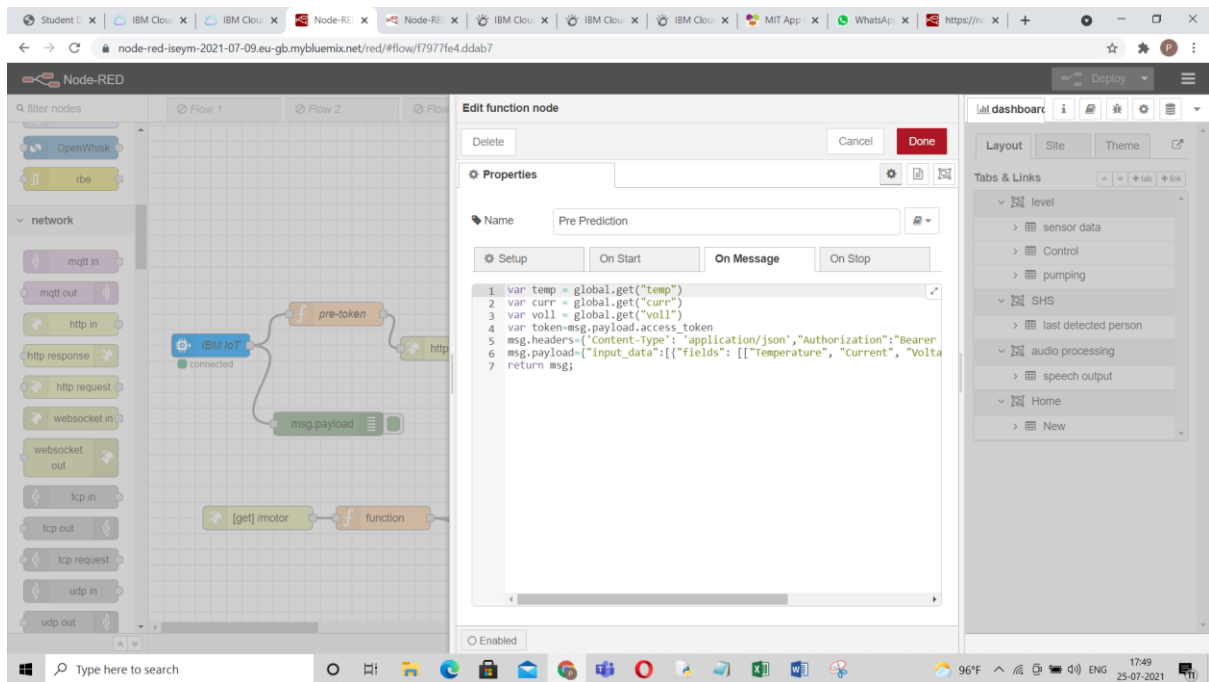
Name	Status	Model type	Last modified
WORKING	Completed	Binary Classification	Jul 25, 2021, 11:33 AM

We need to select the model from the created pipelines, after that we need to get the end point URL for the model which we have deployed

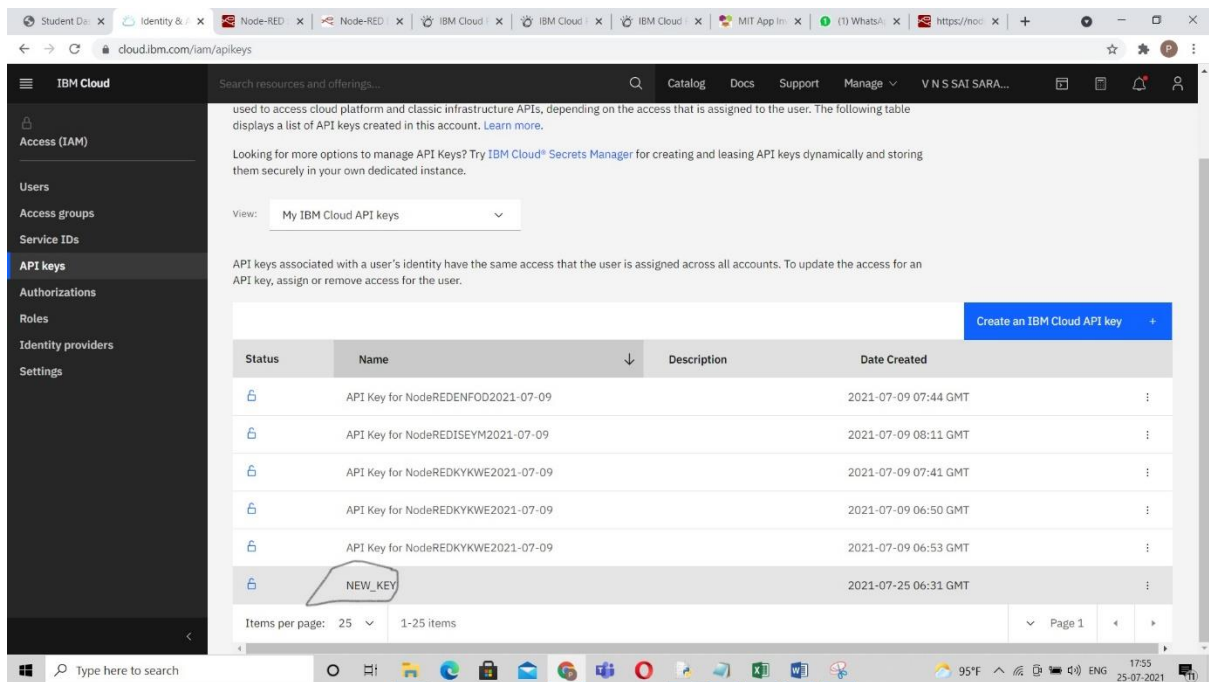
The screenshot shows the IBM Cloud Pak for Data console. The main section is titled "NEW_DEPLOYMENT" and is marked as "Deployed" and "Online". It provides a "Direct link" to the endpoint: `https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/71ad35bf-d622-4aac-acda-9213b7996047/prediction?space_id=ee89f6d1-75ed-479d-9b33-dbad83f7d59d&context=cpsdaas&flush=...`. Below this, there are "Code snippets" for cURL, Java, JavaScript, Python, and Scala. The cURL snippet includes a note about setting the API key and a TODO to manually define values. A sidebar on the right shows details for the deployment, including the creation and update timestamps (Jul 25, 2021 11:53 AM), the deployment ID (71ad35bf-d622-4aac-acda-9213b7996047), the software specification (hybrid_01), and the associated asset (WORKING - P4 Snap Random For...). A pop-up notification for "AutoAI" is also visible.

To develop a node-red application we need to get an "JSON" file to run after getting that json file into the node red, just fill the required details which are needed

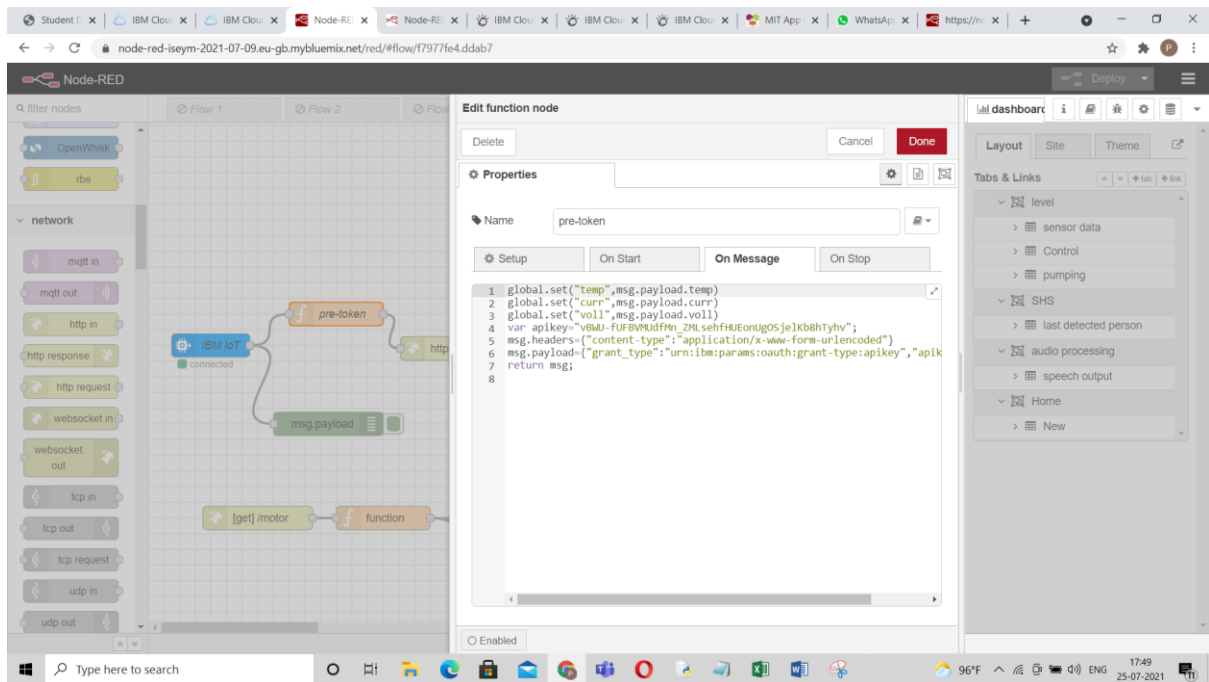




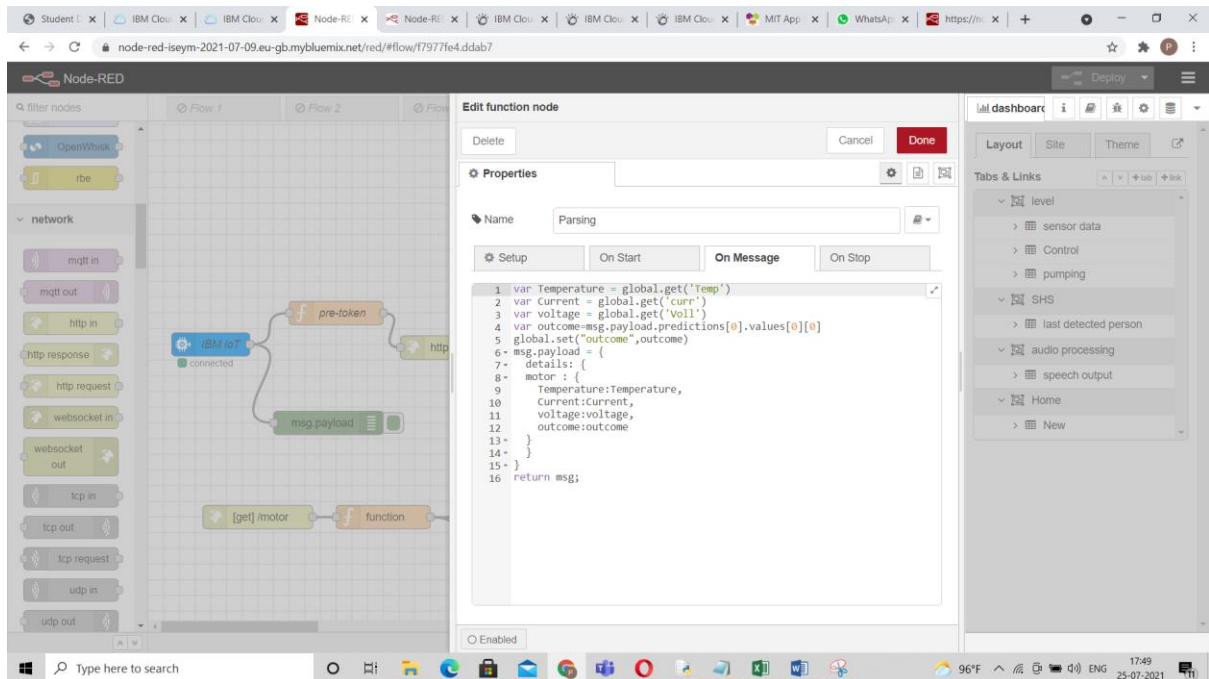
After setting the function we need to generate API-KEY

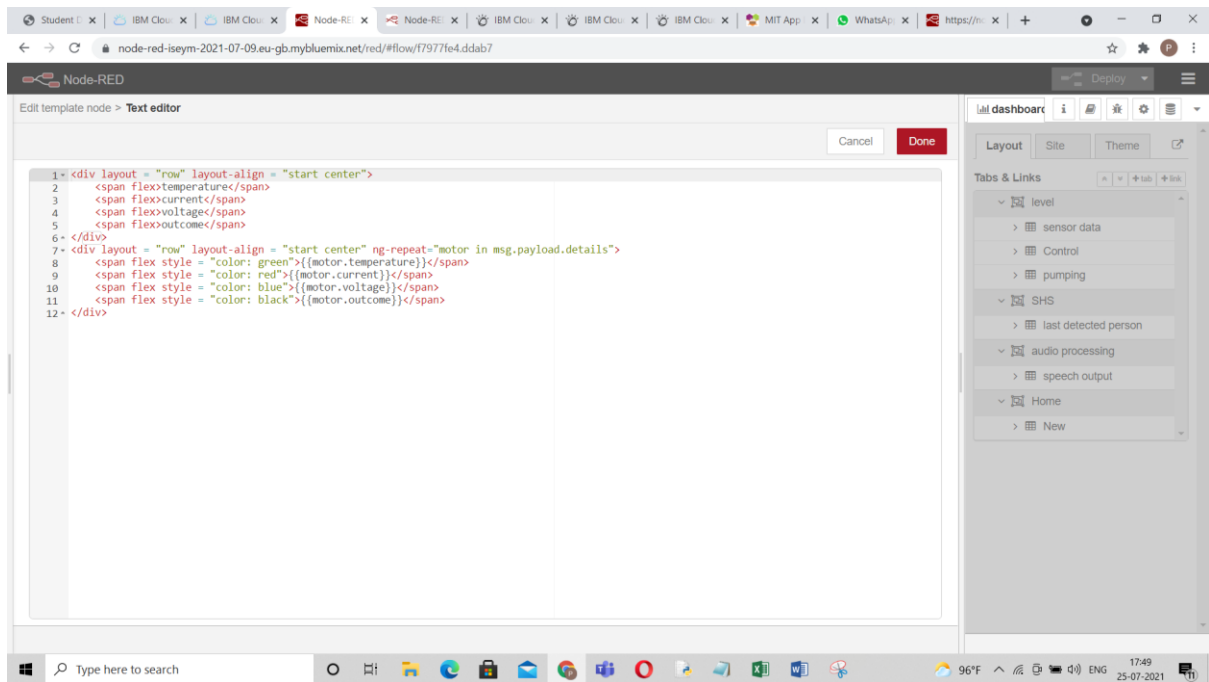


Giving that API-KEY in the function and create that function

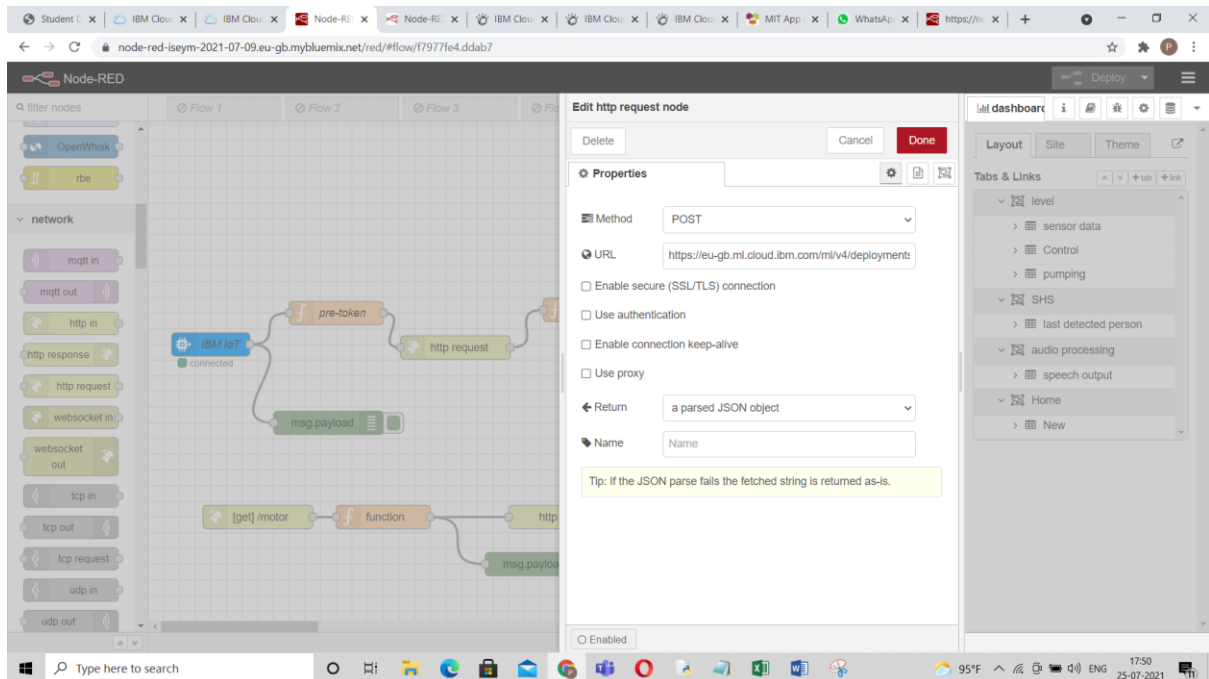


Giving some commands in the functions:

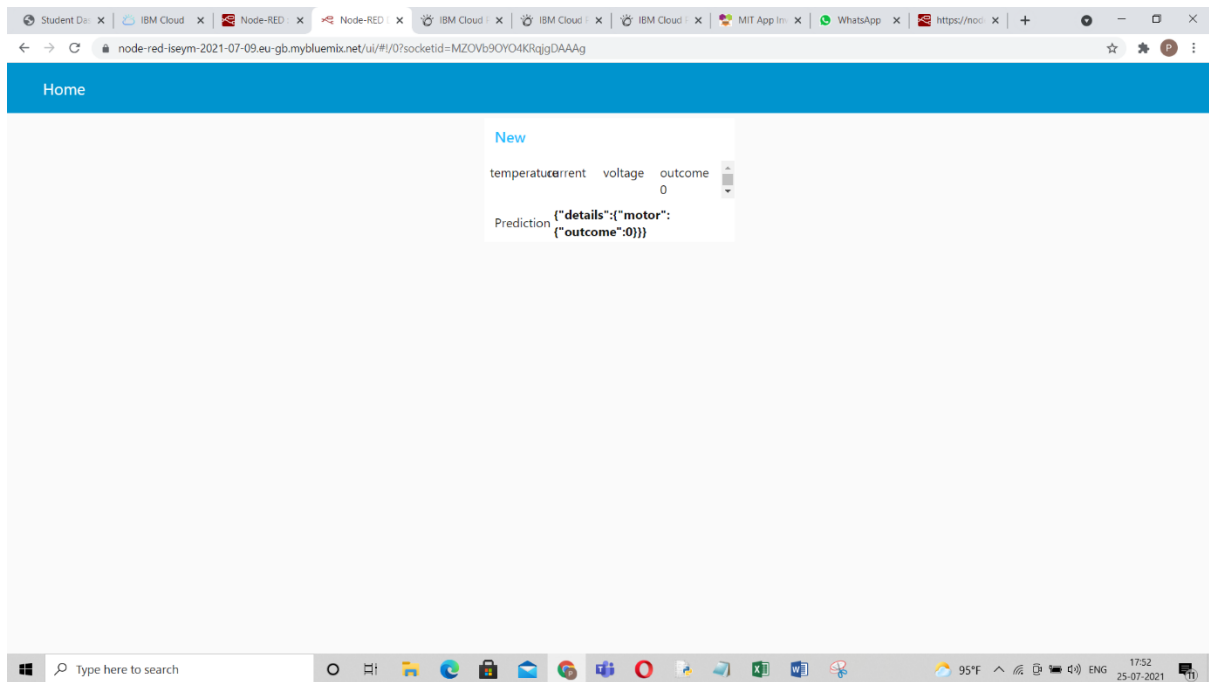




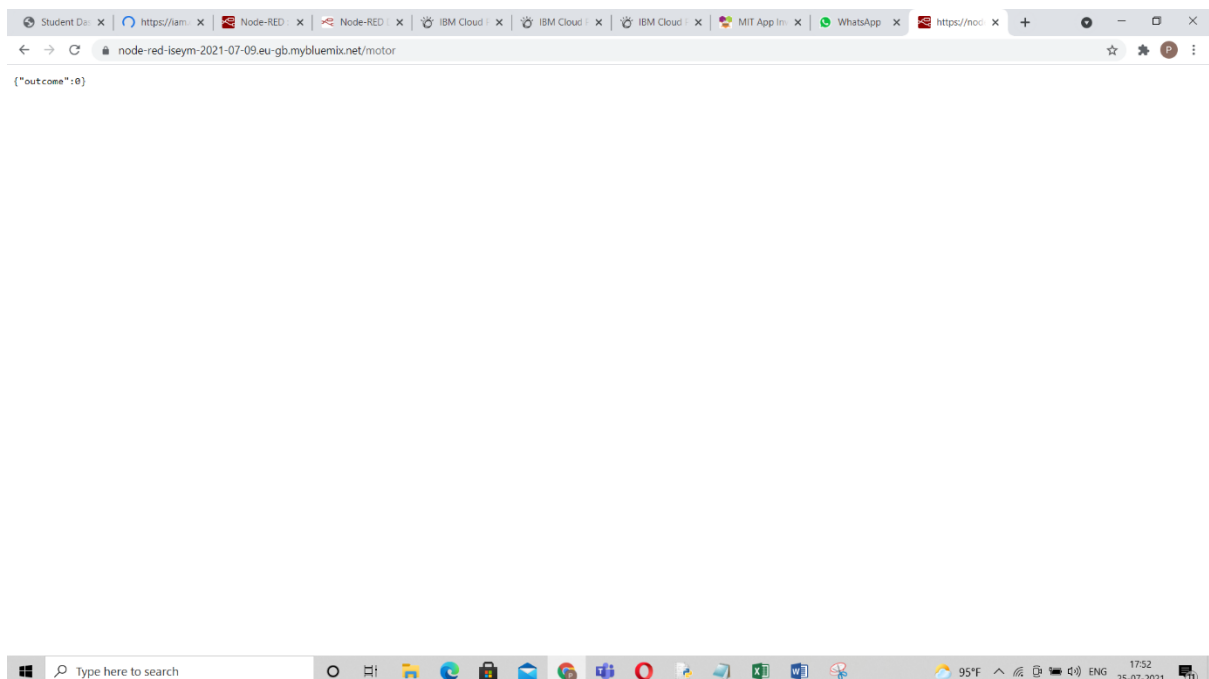
We need to give that end point URL at the one http request node

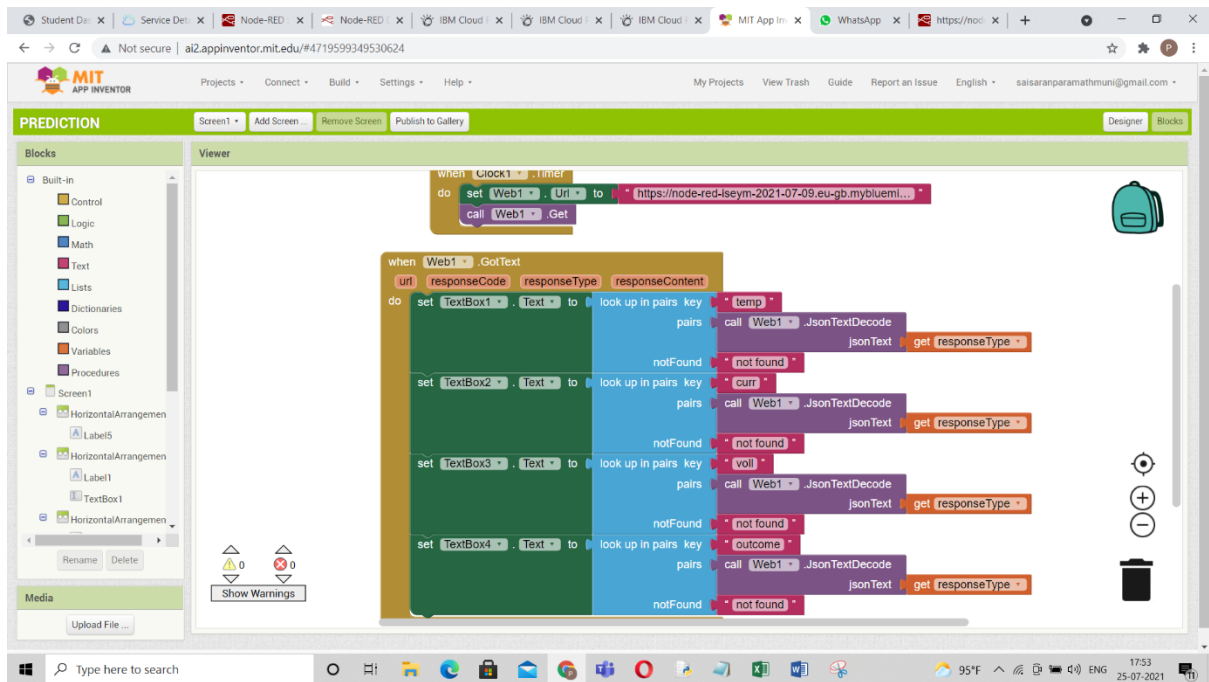
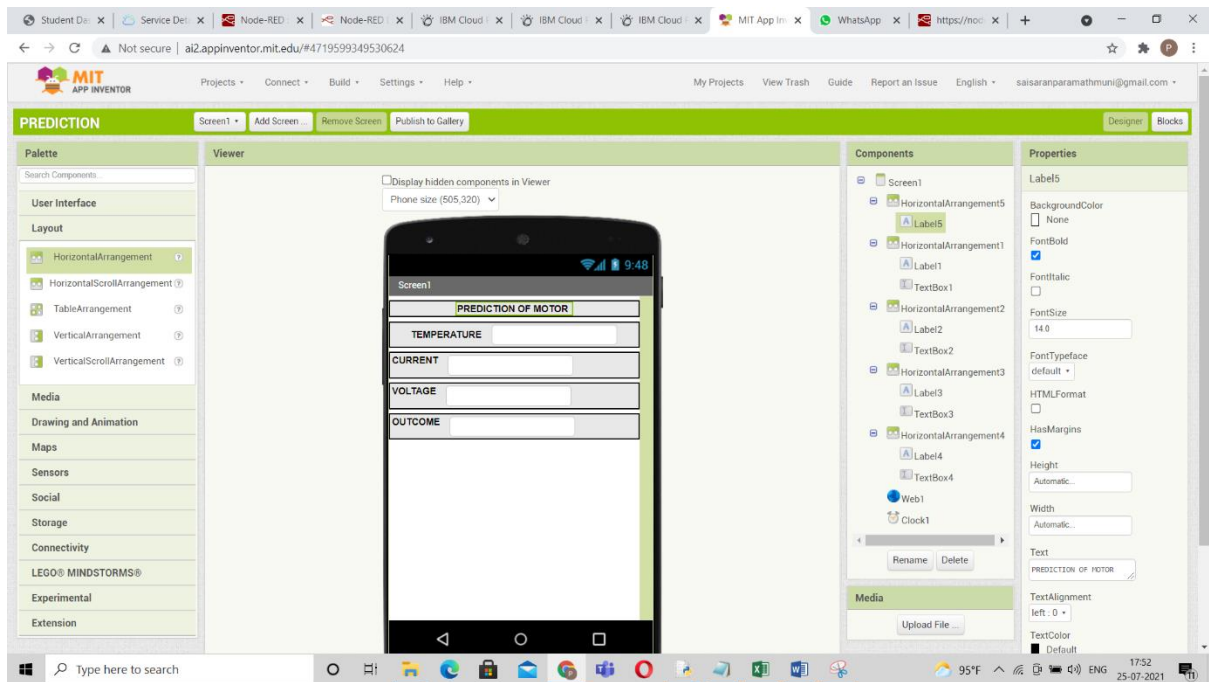


Before deploying the schematic we need to run the python code and after that deploy the schematic we need to view (user interface) the web view of the application

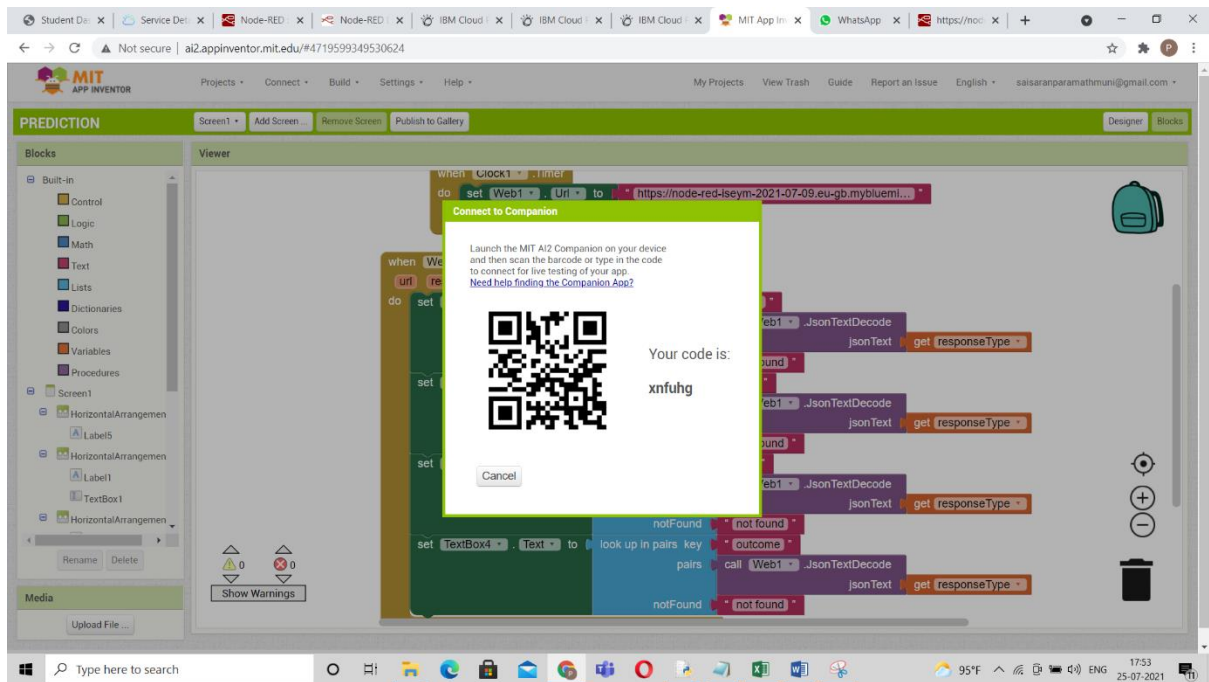


After setting this we need to develop for the app view for that we need get some connections with the same “http/in “ & “http/response” node with including some functions and to develop an app we are using the help of MIT APP INVERTER





After creating the interface and back-end of the app we need to scan and view it in the app in the phone



https://github.com/gnaneshwarbandari/IOT/blob/main/ibm_code.p

The screenshot shows a mobile application interface titled "PREDICTION OF MOTOR". At the top, there is a status bar with the time 4:33, battery level 69%, and various icons. Below the status bar, the app title "PREDICTION OF MOTOR" is displayed. The interface contains four input fields with labels to their left: "TEMPERATURE" with a hint "Hint for TextBox1", "CURRENT" with a hint "Hint for TextBox2", "VOLTAGE" with a hint "Hint for TextBox3", and "OUTCOME" which displays the value "0". The app is running on a device with a green status bar and a grey home indicator bar at the bottom.

The user will have a connection to this app so he can view it

ADVANTAGES & DISADVANTAGES:

Reduction in maintenance costs

Reduction in machine failures

Life time of service parts will increase

Safety operator

Increases investment in diagnostic equipment

Savings potential is readily seen by management

Increases investment in staff training

CONCLUSION:

This project is mainly focused on the problem of carrying out predictive maintenance in a industrial motors and presented the results of the preliminary data analysis and feature selection that were performed on a sample of the collected data sheet. The derived data from IOT device gives the status of industrial motors about temperature, current & voltage from the equipment is continuously monitored to avoid any short circuits and line breakage. So predictive maintenance of industrial motors plays a major roll in maintaining it. In order to reduce the risk factor in this the process has to be carefully planned and carried out by well trained workers

0 – it won't work

1 – it will work

BIBLIOGRAPHY:

Smart bridge lecture videos

IBM platform videos

<https://drive.google.com/file/d/1W4skAVwWkVhKBbSMAMXtnrvP6MbdqvtG/view?usp=sharing>

<https://drive.google.com/file/d/1tX2MhuhqvMfHpirDIRhpSODV1gFtXqL9/view?usp=sharing>

https://drive.google.com/file/d/1oL_KpIKNeSuwVNnsJhQ6rgY3MIRogHvS/view?usp=sharing

SOURCE CODE:

https://github.com/gnaneshwarbandari/IOT/blob/main/ibm_code.py