SMART BRIDGE : PROJECT

Smart security and safety solutions based on IoT for large industrial plants

Team Members:

1. K. Harshith
   kanuthala.harshith2019@vitstudent.ac.in
2. Y. Sushwanth Reddy
   yanamalasushwanth.2019@vitstudent.ac.in
3. N. Varun Krishna
   varun.krishna2019@vitstudent.ac.in
4. K. Varun Reddy
   kunnamvarun.reddy2019@vitstudent.ac.in

Introduction:

Overview:

To create smart security and safety system for large industrial plants.

Objective:

To develop smart security and safety system which stores the date and time of employee while entry and exit using IBM cloud. Alerts the employee who exceeds the time duration in radiation room by using buzzer and by displaying message to leave the room in OLED screen.

Survey:

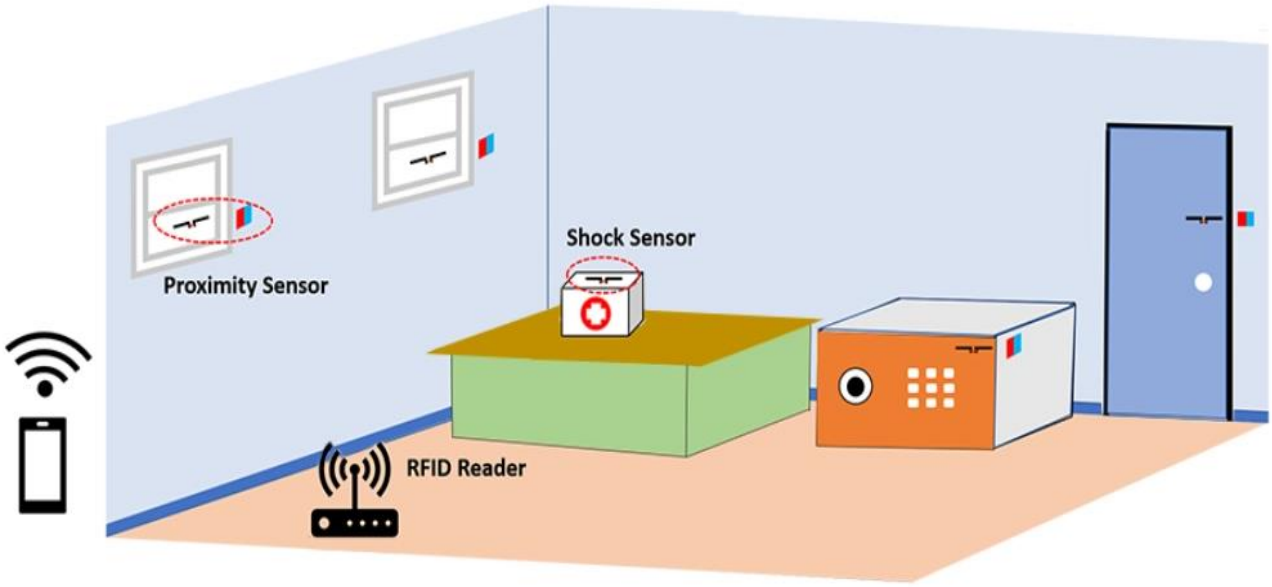
Existing model:

General model which can store date and time for entry and exit.

Solution:

We can calculate the total time spent in company using IBM cloud, we can send alert to employee in radiation room who exceeds the time limit. This can save the employee from radiation effect.

Block Diagram:



Requirements:

Python IDLE

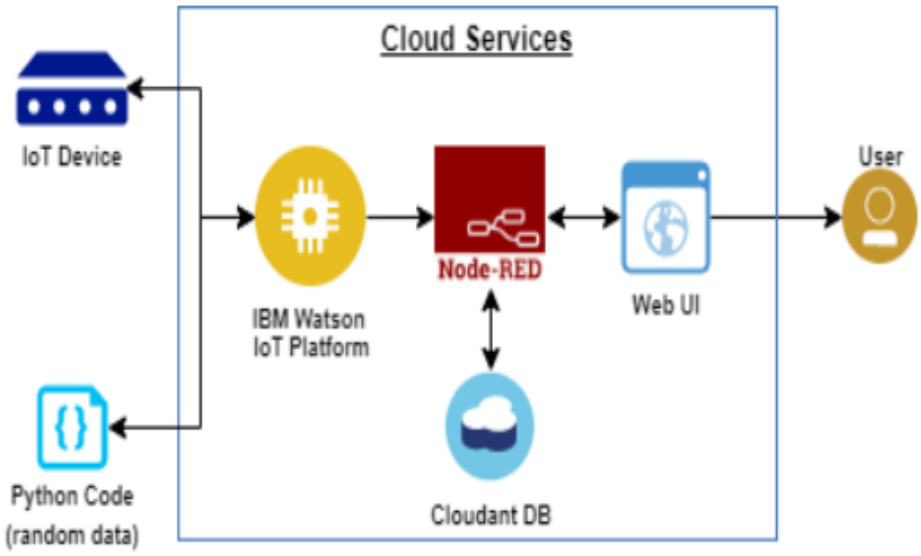
IBM account:

IBM Cloud

IBM Watson IoT platform

Node-red

Flow Chart:



Procedure:

Develop the code:

IOTPROJECT.py - C:/Users/NITHIN/AppData/Local/Programs/Python/Python39/IOTPROJECT.py (3.9.6)

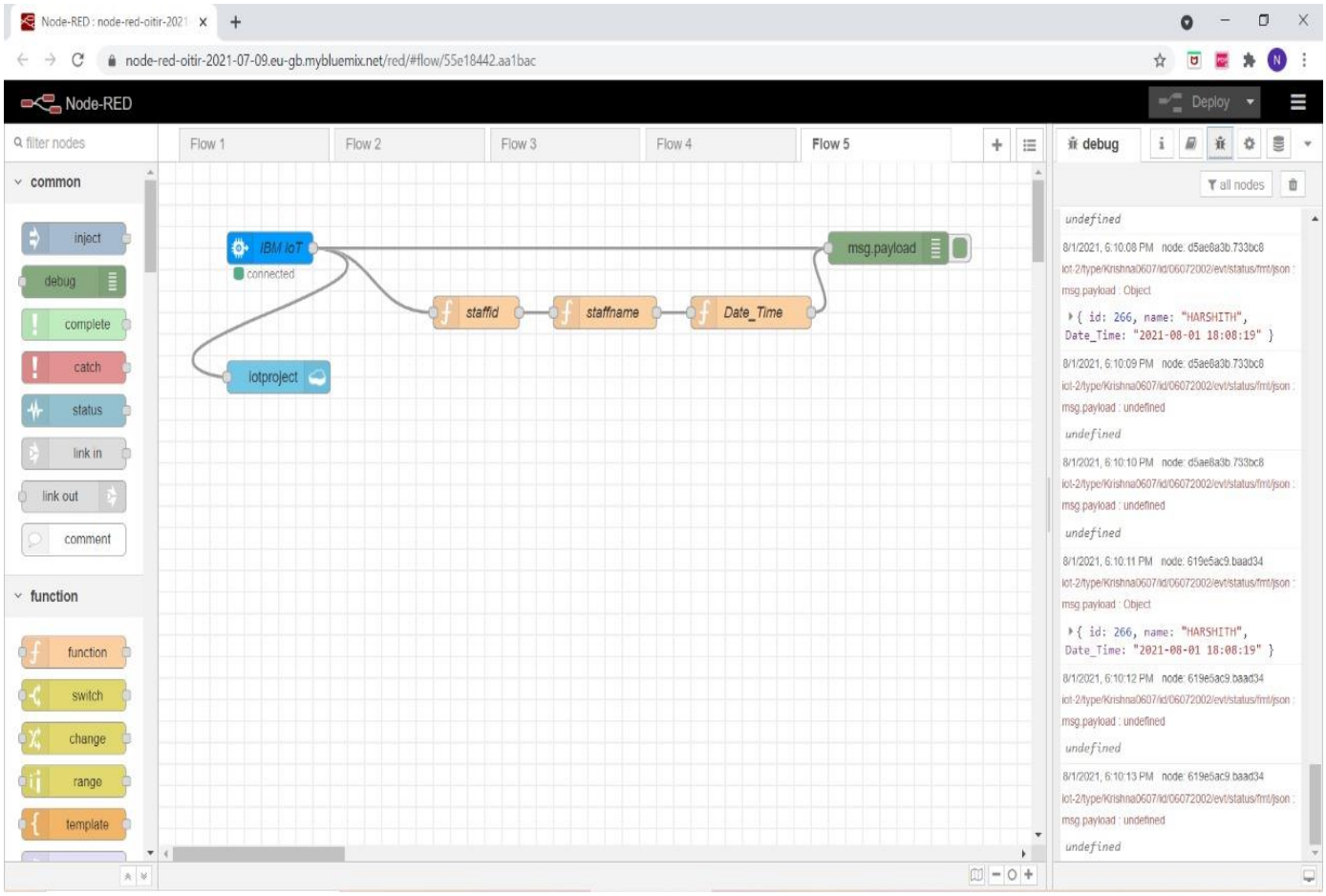
File Edit Format Run Options Window Help

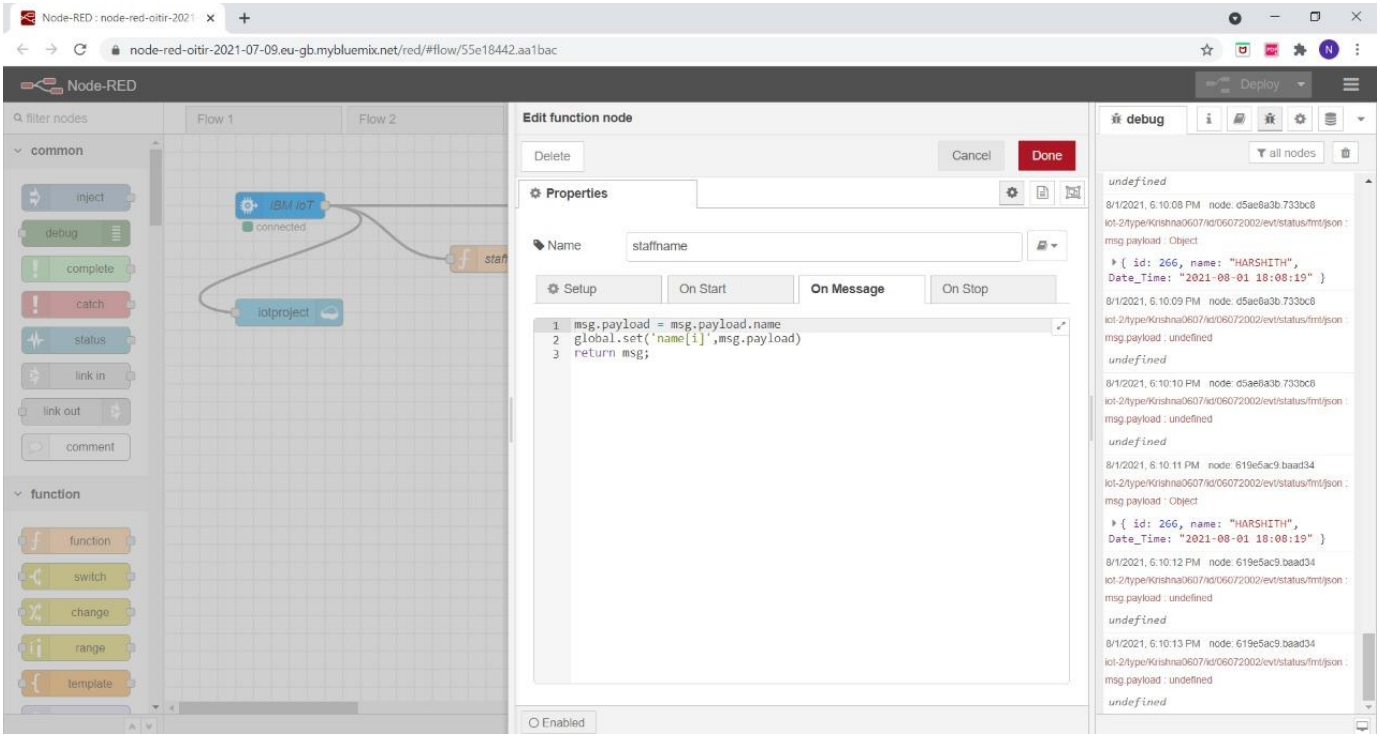
```
import wiotp.sdk.device
import datetime
import time
import random
myConfig = {
    "identity": {
        "orgId": "xmf5vy",
        "typeId": "Y1",
        "deviceId":"9399996166"
    },
    "auth": {
        "token": "9399996166"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

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

name = { 273: 'VARUN', 266: 'HARSHITH', 272: 'SUSWANTH', 279: 'KUNNAM'}
now=datetime.datetime.now()
date_time = now.strftime("%Y-%m-%d %H:%M:%S")


while True:

    def Enter_Industry():
        i=int(input("Enter the ID to Enter Industry Plant: "))
        myData={'id':i, 'name':name[i], 'Date_Time':date_time }
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published data Succesfully: %s",myData)
        print( name[i]+" Entered Industry Plant")

    def Enter_Radiation():
        j=int(input("Enter the ID to Enter Radiation room: "))
        myData={'id':j, 'name':name[j], 'Date_Time':date_time }
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published data Succesfully: %s",myData)
        print( name[j]+" Entered Radiation room")
```

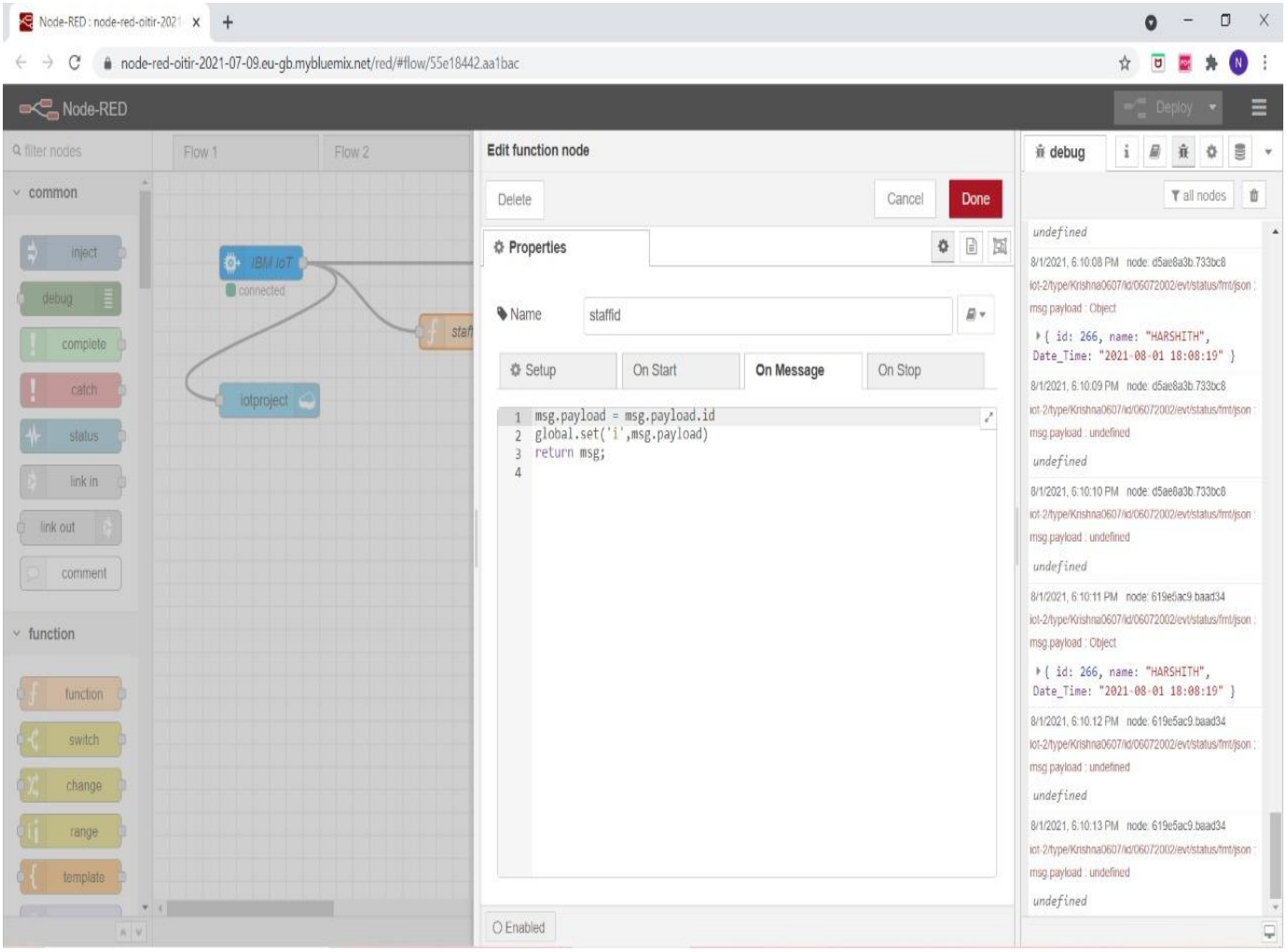
```
    def Exit_Radiation():
        k=int(input("Enter the ID to Exit Radiation room: "))
        myData={'id':k, 'name':name[k], 'Date_Time':date_time }
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published data Succesfully: %s",myData)
        print( name[k]+" Exited Radiation room")

    def Exit_Industy():
        l=int(input("Enter the ID to Exited Indusry Plant: "))
        myData={'id':l, 'name':name[l], 'Date_Time':date_time }
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published data Succesfully: %s",myData)
        print( name[l]+" Exited Industry Plant")
    def default():
        print("Selected Choice is not correct")

    switcher = {
        1: Enter_Indusrty,
        2: Enter_Radiation,
        3: Exit_Radiation,
        4: Exit_Industy,
        }

    def switch(operation):
        return switcher.get(operation, default)()

    print('''You can perform operation
    1. Enter Industry plant
    2. Enter Radiation room
    3. Exit Radiation room
    4. Exit Industry plant ''')

    #Take input from user
    choice = int(input("Select operation from 1,2,3,4 : "))
    print (switch(choice))



    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
{"mode":"full","isActive":flase}
```

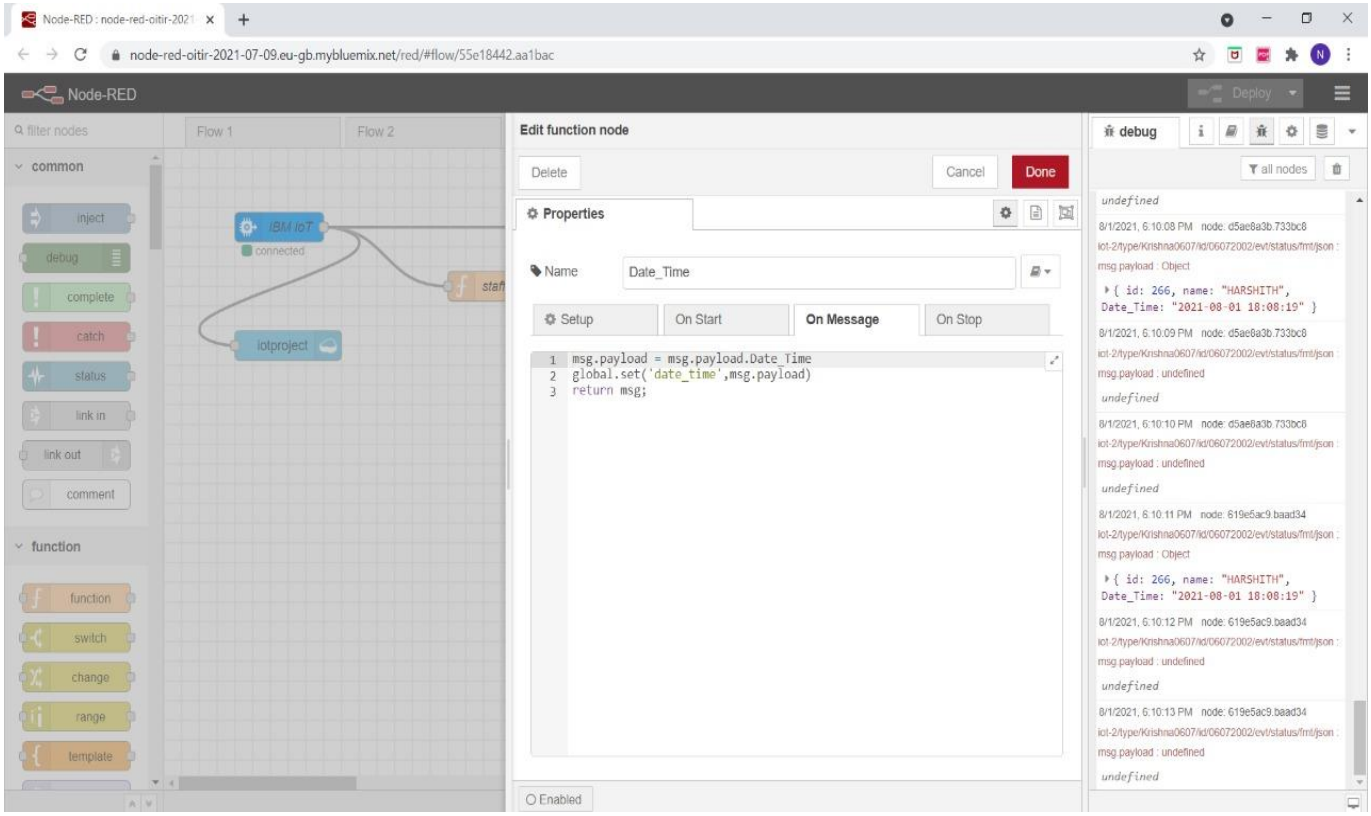
Ln: 7 Col: 26

Node-red:

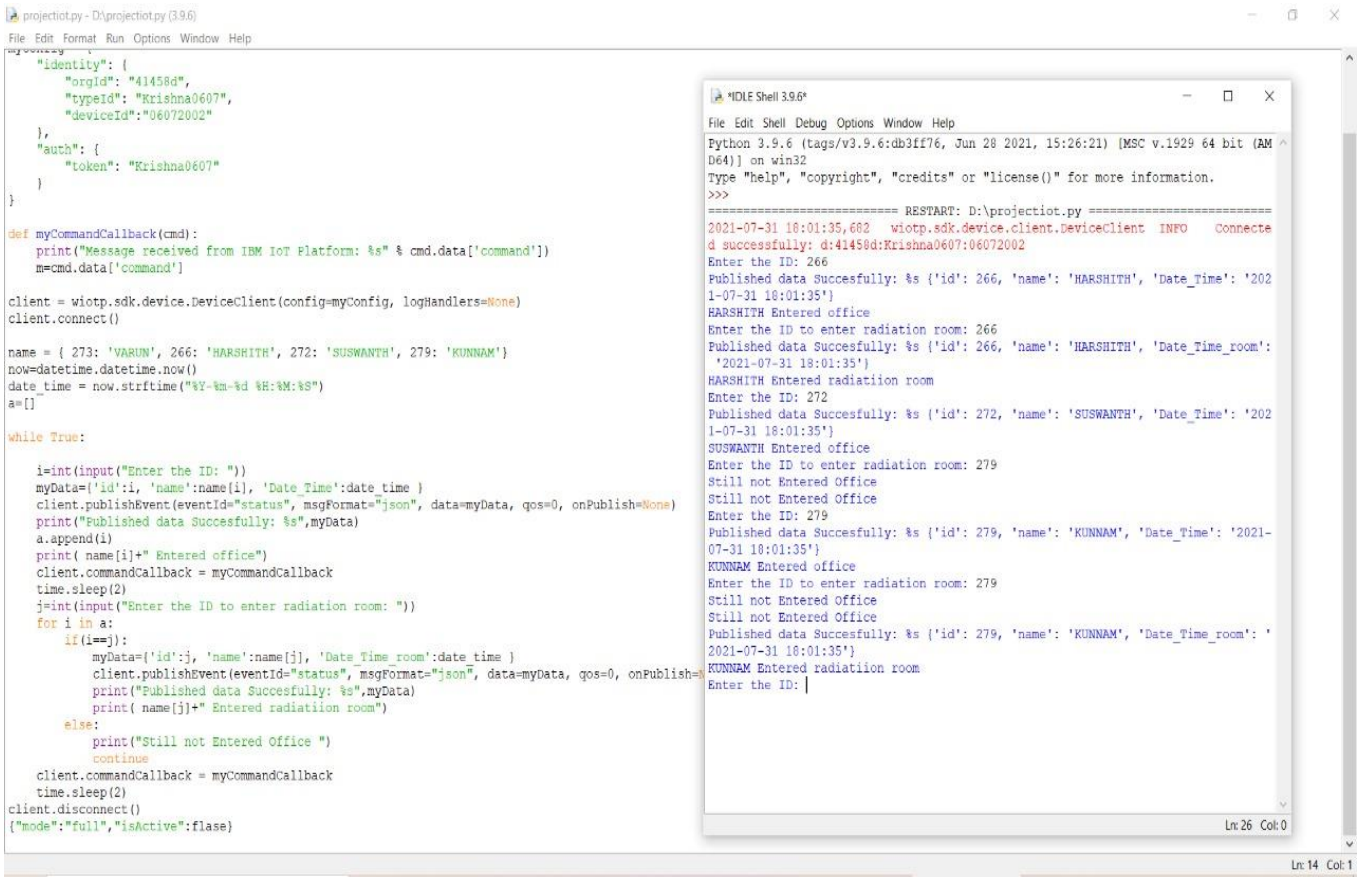


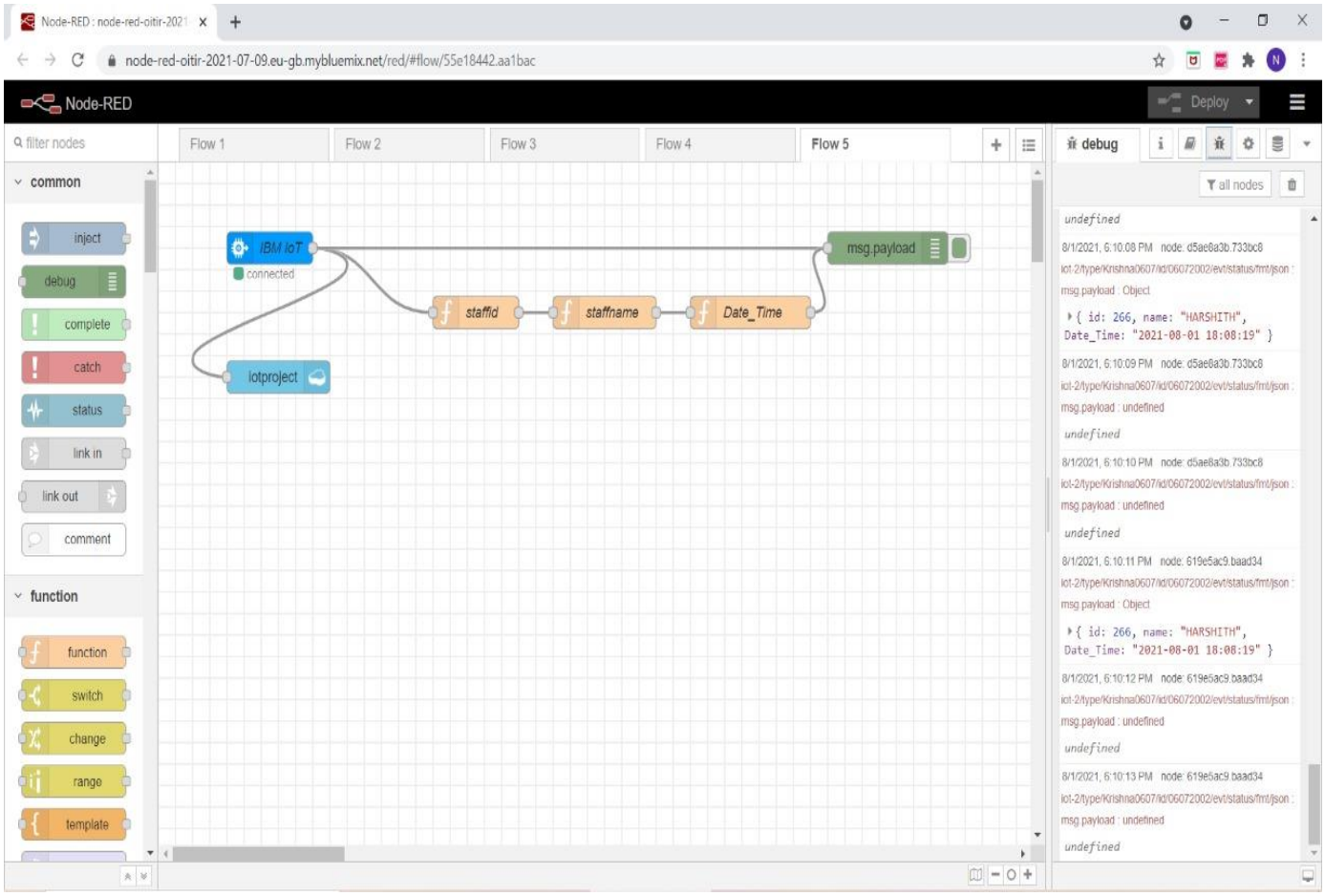

Node-RED : node-red-oitir-2021
node-red-oitir-2021-07-09.eu-gb.mybluemix.net/red/#flow/55e18442.aa1bac
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
iotproject
Edit function node
Delete
Cancel
Done
Properties
Name
staffname
Setup
On Start
On Message
On Stop
1 msg.payload = msg.payload.name
2 global.set('name[i]',msg.payload)
3 return msg;
Enabled
debug
all nodes
undefined
8/1/2021, 6:10:08 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:09 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:10 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:11 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:12 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:13 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined


Node-RED : node-red-oitir-2021
node-red-oitir-2021-07-09.eu-gb.mybluemix.net/red/#flow/55e18442.aa1bac
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
iotproject
Edit function node
Delete
Cancel
Done
Properties
Name
staffid
Setup
On Start
On Message
On Stop
1 msg.payload = msg.payload.id
2 global.set('i',msg.payload)
3 return msg;
4
Enabled
debug
all nodes
undefined
8/1/2021, 6:10:08 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:09 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:10 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:11 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:12 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:13 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined


Node-RED : node-red-oitir-2021
node-red-oitir-2021-07-09.eu-gb.mybluemix.net/red/#flow/55e18442.aa1bac
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
staff
Iotproject
Edit function node
Delete
Cancel
Done
Properties
Name
Date_Time
Setup
On Start
On Message
On Stop
msg.payload = msg.payload.Date_Time
global.set('date_time',msg.payload)
return msg;
Enabled
debug
all nodes
undefined
8/1/2021, 6:10:08 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:09 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:10 PM node: d5ae8a3b.733bc8
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:11 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : Object
{ id: 266, name: "HARSHITH", Date_Time: "2021-08-01 18:08:19" }
8/1/2021, 6:10:12 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined
8/1/2021, 6:10:13 PM node: 619e5ac9.baad34
iot-2/type/Krishna0607/id/06072002/evt/status/fmt/json : msg.payload : undefined
undefined

Result:


projectiot.py - D:\projectiot.py (3.9.6)
File Edit Format Run Options Window Help
"identity": {
"orgId": "41458d",
"typeId": "Krishna0607",
"deviceId":"06072002"
},
"auth": {
"token": "Krishna0607"
}
}
def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
name = { 273: 'VARUN', 266: 'HARSHITH', 272: 'SUSWANTH', 279: 'KUNNAM'}
now=datetime.datetime.now()
date_time = now.strftime("%Y-%m-%d %H:%M:%S")
a=[]
while True:
i=int(input("Enter the ID: "))
myData={'id':i, 'name':name[i], 'Date_Time':date_time }
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Published data Succesfully: %s",myData)
a.append(i)
print(name[i]+" Entered office")
client.commandCallback = myCommandCallback
time.sleep(2)
j=int(input("Enter the ID to enter radiation room: "))
for i in a:
if(i==j):
myData={'id':j, 'name':name[j], 'Date_Time_room':date_time }
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=
print("Published data Succesfully: %s",myData)
print(name[j]+" Entered radiatiion room")
else:
print("Still not Entered Office ")
continue
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
{"mode":"full","isActive":flase}
Ln: 14 Col: 1
IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\projectiot.py ==========
2021-07-31 18:01:35,682 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:41458d:Krishna0607:06072002
Enter the ID: 266
Published data Succesfully: %s {'id': 266, 'name': 'HARSHITH', 'Date_Time': '2021-07-31 18:01:35'}
HARSHITH Entered office
Enter the ID to enter radiation room: 266
Published data Succesfully: %s {'id': 266, 'name': 'HARSHITH', 'Date_Time_room': '2021-07-31 18:01:35'}
HARSHITH Entered radiatiion room
Enter the ID: 272
Published data Succesfully: %s {'id': 272, 'name': 'SUSWANTH', 'Date_Time': '2021-07-31 18:01:35'}
SUSWANTH Entered office
Enter the ID to enter radiation room: 279
Still not Entered Office
Still not Entered Office
Enter the ID: 279
Published data Succesfully: %s {'id': 279, 'name': 'KUNNAM', 'Date_Time': '2021-07-31 18:01:35'}
KUNNAM Entered office
Enter the ID to enter radiation room: 279
Still not Entered Office
Still not Entered Office
Published data Succesfully: %s {'id': 279, 'name': 'KUNNAM', 'Date_Time_room': '2021-07-31 18:01:35'}
KUNNAM Entered radiatiion room
Enter the ID:
Ln: 26 Col: 0



Advantages:

Reduction in maintenance costs

Safety operator

Secured data

Flexible

Easy accessibility

Interference

Disadvantages:

Cost

High storage required

Disturbances

Conclusion:

This project is mainly focused on date and time of entry/exit of the employee. This can measure the total time spent in company by calculating time from exit and entry time. To ensure the safety of the employee, we use countdown from the time of entry to the radiation room and when the time limit has reached, alert is given to employee by making the buzzer on and displaying message to leave the radiation room in the OLED screen. This helps the safety of the employee. All the data is stored in the cloud, which can be accessed easily.

Bibolography:

Smart bridge lecture videos

IBM platform videos

https://thesmartbridge.com/documents/projects/SmartHomeAutomationusingIBMCloud.pdf

Source code:

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

https://github.com/divyanemuri/SmartInternz-IoT-Externship-2021/blob/master/Python_pubsubIBM.py