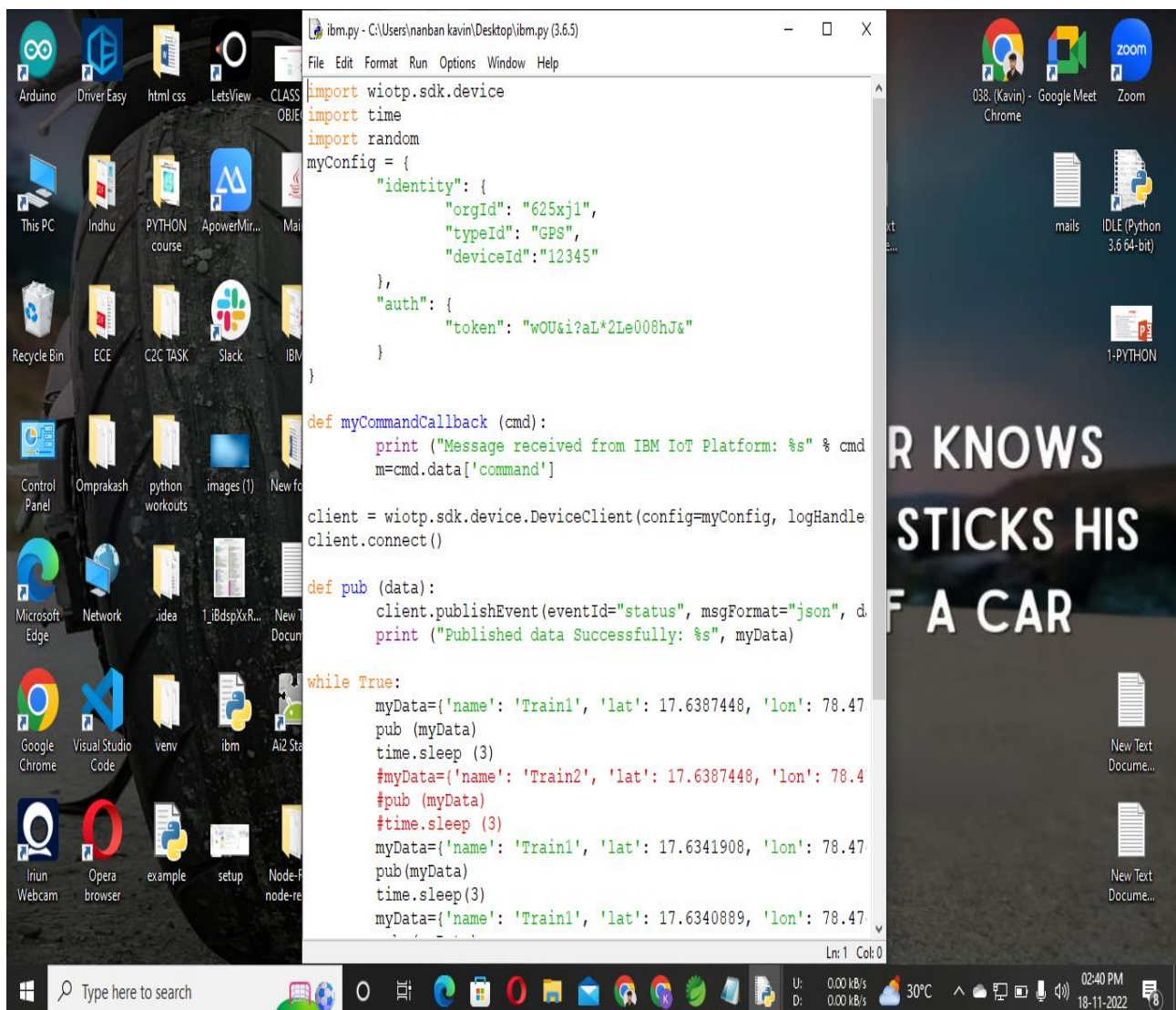


SPRINT 3

DEVELOPED PYTHON SCRIPT

Date	13 NOVEMBER 2022
Team ID	PNT2022TMID13493
Title	SMART SOLUTION FOR RAILWAYS



The screenshot shows a Windows 10 desktop environment. In the center, a code editor window titled 'ibm.py - C:\Users\nanban kavin\Desktop\ibm.py (3.6.5)' is open. The script is a Python program that interacts with the IBM IoT Platform. It imports 'wiotp.sdk.device', 'time', and 'random'. A configuration dictionary 'myConfig' is defined with 'identity' (orgId: '625xj1', typeId: 'GPS', deviceId: '12345') and 'auth' (token: 'wOU&i?al*2Le008hJ&'). A function 'myCommandCallback' prints messages received from the IBM IoT Platform. The main logic is in a 'while True' loop that publishes data to the platform. The data is a dictionary with 'name', 'lat', and 'lon' fields. The script publishes data for 'Train1' and 'Train2' with specific coordinates, followed by a 3-second sleep. The desktop background is a dark image with the text 'NO ONE KNOWS WHAT HE KNOWS STICKS HIS FINGER IN THE MOUTH OF A CAR'. Various application icons are visible on the desktop, including Arduino, Driver Easy, html css, LetsView, CLASS OBJECT, This PC, Indhu, PYTHON course, ApowerMir..., Mail, Recycle Bin, ECE, C2C TASK, Slack, IBM, Control Panel, Omprakash, python workouts, images (1), New folder, Microsoft Edge, Network, idea, 1_ibdspXr..., New Document, Google Chrome, Visual Studio Code, venv, ibm, Ai2 Sta..., Injun Webcam, Opera browser, example, setup, and Node-F node-re. The taskbar at the bottom shows the Start button, a search bar, and several open applications. The system tray on the right indicates the time as 02:40 PM on 18-11-2022, with system icons for network, volume, and battery.

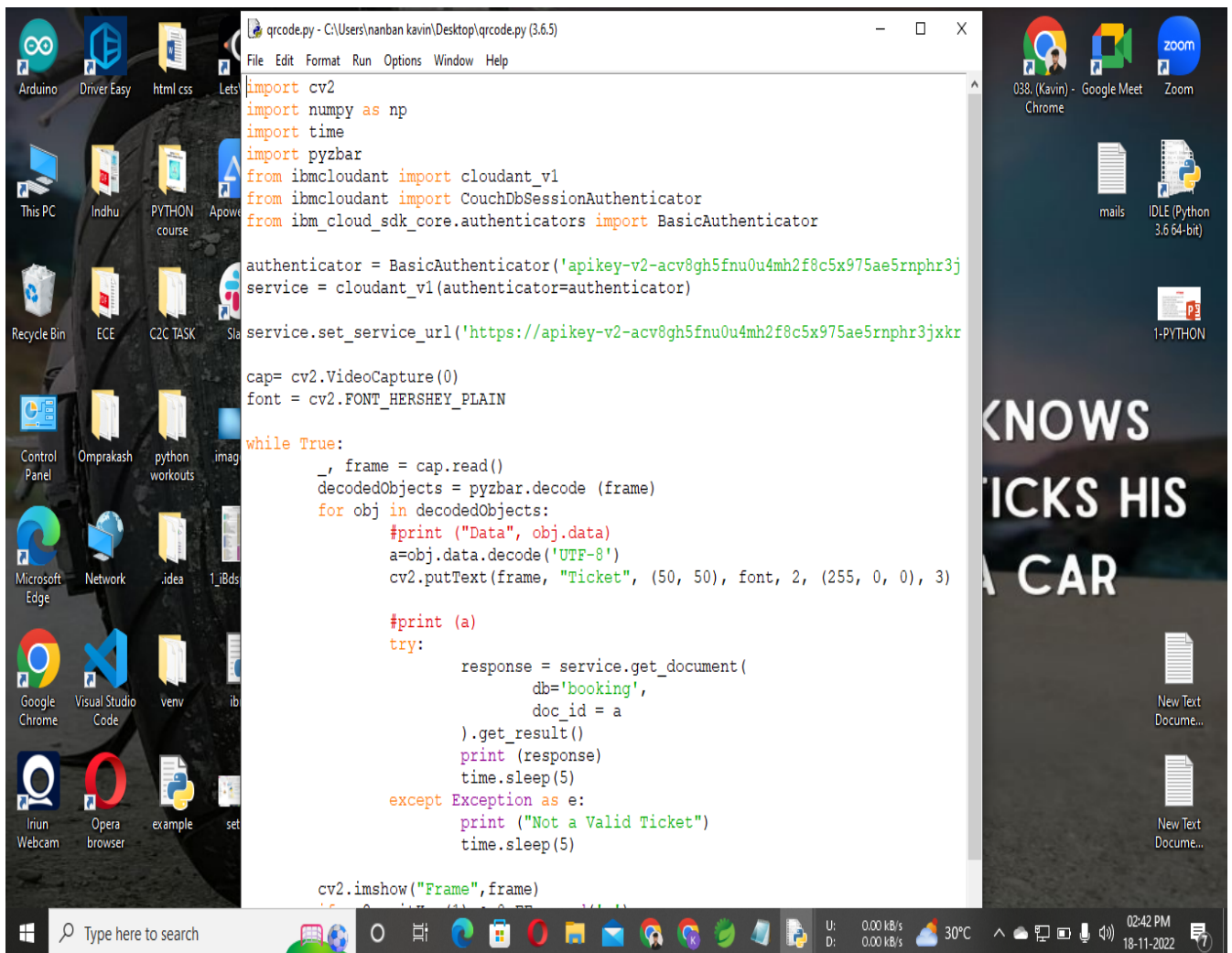
```
ibm.py - C:\Users\nanban kavin\Desktop\ibm.py (3.6.5)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "625xj1",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "wOU&i?al*2Le008hJ&"
    }
}

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

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandle
client.connect()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="json", d
    print ("Published data Successfully: %s", myData)

while True:
    myData={'name': 'Train1', 'lat': 17.6387448, 'lon': 78.47
    pub (myData)
    time.sleep (3)
    #myData={'name': 'Train2', 'lat': 17.6387448, 'lon': 78.4
    #pub (myData)
    #time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6341908, 'lon': 78.47
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6340889, 'lon': 78.47
```



```
*Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
Ln: 54 Col: 0
```

IBM:

```
import wiotp.sdk.device

import time
import random
myConfig = {
    "identity": {
        "orgId": "625xj1",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "wOU&i?aL*2Le008hJ&"
    }
}

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()
```

```
def pub (data):
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
print ("Published data Successfully: %s", myData)
```

```
while True:
myData={'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
pub (myData)
time.sleep (3)
#myData={'name': 'Train2', 'lat': 17.6387448, 'lon': 78.4754336}
#pub (myData)
#time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
pub(myData)
time.sleep(3)
myData={'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
pub (myData)
time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
pub (myData)
time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
pub (myData)
time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
pub (myData)
time.sleep (3)
client.commandCallback = myCommandCallback
client.disconnect ()
```

QR CODE:-

```
import cv2
import numpy as np
import time
import pyzbar
from ibmcloudant import cloudant_v1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-
acv8gh5fnu0u4mh2f8c5x975ae5rnphr3jxkr5d9ril','c1dd4db6e976d915751882f688e410ec')
service = cloudant_v1(authenticator=authenticator)
```

```
service.set_service_url('https://apikey-v2-  
acv8gh5fnu0u4mh2f8c5x975ae5rnphr3jxkr5d9ril:c1dd4db6e976d915751882f688e410ec@a  
dad2af9-59c4-41bb-b4b4-806f0d6962b2-bluemix.cloudantnosqldb.appdomain.cloud')
```

```
cap= cv2.VideoCapture(0)  
font = cv2.FONT_HERSHEY_PLAIN
```

```
while True:  
    __, frame = cap.read()  
    decodedObjects = pyzbar.decode (frame)  
    for obj in decodedObjects:  
        #print ("Data", obj.data)  
        a=obj.data.decode('UTF-8')  
        cv2.putText(frame, "Ticket", (50, 50), font, 2, (255, 0, 0), 3)
```

```
#print (a)  
try:  
    response = service.get_document(  
        db='booking',  
        doc_id = a  
    ).get_result()  
    print (response)  
    time.sleep(5)  
except Exception as e:  
    print ("Not a Valid Ticket")  
    time.sleep(5)
```

```
cv2.imshow("Frame",frame)  
if cv2.waitKey(1) & 0xFF ==ord('q'):  
    break  
cap.release()  
cv2.destroyAllWindows()  
client.disconnect()
```

