

## SPRINT 3

<b>Project name</b>	Smart solutions for Railways
<b>Project id</b>	PNT2022TMID38437

### Team members:

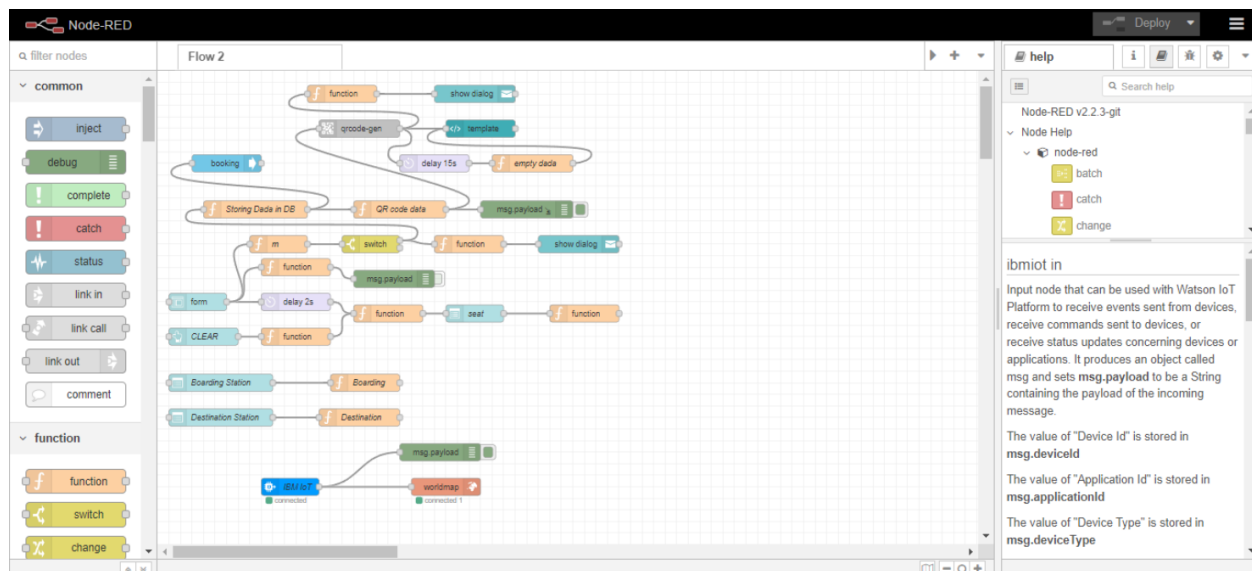
Aarthi.A(413019104001)

Abinaya.c(413019104002)

Bhuvaneshwari.M(413019104007)

Deepapriya.S(413019104009)

Developer web application:node red program



### Qr code generation:

```
import cv2
import numpy as np
import time
import pyzbar . pyzbar as puzbar
from ibmcloudant . cloudant_v1 import cloudantv1
```

```

from ibmcloudant import couchDbSessionAuthenticator
from ibm_cloud_sdk_core.Authenticators import BasicAuthenticator

authenticator=BasicAuthenticator('apikey-v2-
16u3crmdpkgghxefdikvpssoh5fwezrmuup5fv5g3ubz','b0ab119f45d3e6255eabb978')
service =cloudantv1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-
16u3crmdpkgghxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119f45d3e6255eabb978')
cap = cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_PLAIN

    while True:
        _, frame = cap.read(0)
        decodeObjects = pyzbar.decode(frame)
        for obj in decodeObjects:
            #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 valid Ticket")
            time.sleep(5)

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

```

## **Ticket booking program:**

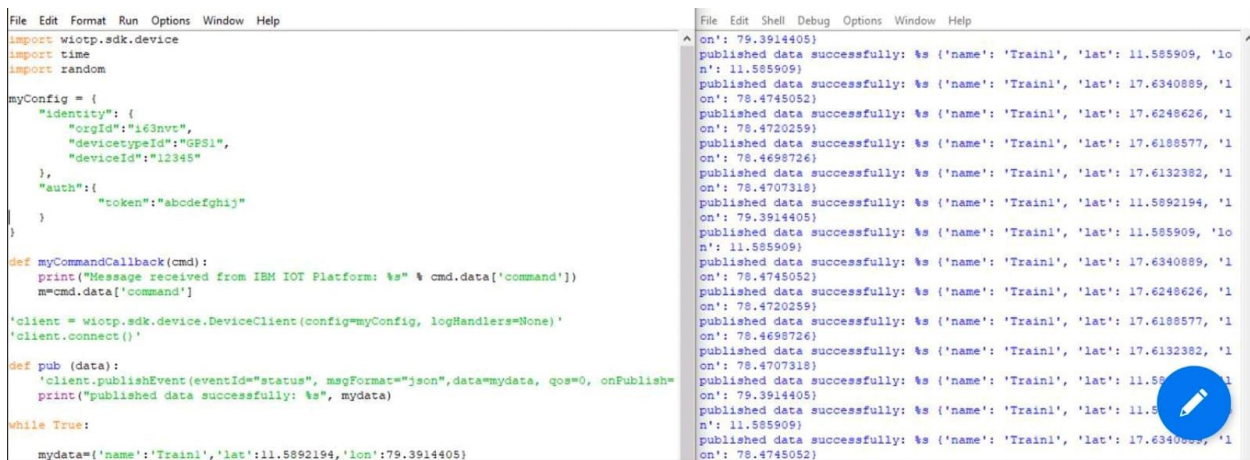
```
1.         import wiotp.sdk.device
2.         import time
3.         import random
4.         myConfig = {
5.             "identity": {
6.                 "orgId":"i63nvt",
7.                 "devicetypeId":"GPS1",
8.                 "deviceId":"i2345"
9.             },
10.            "auth":{
11.                "token":"abcdefghij"
12.            }
13.        }
14.
15.    def myCommandCallback(cmd):
16.        print("Message received from IBM IOT Platform: %s" % cmd.data['command'])
17.        m=cmd.data['command']
18.
19. 'client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)' 20.
'client.connect()'
21.
22.    def pub (data):
23.        'client.publishEvent(eventId="status", msgFormat="json",data=mydata, qos=0,
24.        onPublish=None)'
25.        print("published data successfully: %s", mydata)
26.
27.
28.    mydata={'name':'Train1','lat':17.6387448,'lon': 78.4754336}
29.    pub(mydata)
30.    time.sleep(3)
```

```

31.     #mydata={'name':'Train2','lat':17.6387448,'lon': 78.4754336}
32.     #pub(mydata)
33.     #time.sleep(3)
34.     mydata={'name':'Train1','lat':17.6341908,'lon': 78.4744722}
35.     pub(mydata)
36.     time.sleep(3)
37.     mydata={'name':'Train1','lat':17.6340889,'lon': 78.4745052}
38.     pub(mydata)
39.     time.sleep(3)
40.     mydata={'name':'Train1','lat':17.6248626,'lon': 78.4720259}
41.     pub(mydata)
42.     time.sleep(3)
43.     mydata={'name':'Train1','lat':17.6188577,'lon': 78.4698726}
44.     pub(mydata)
45.     time.sleep(3)
46.     mydata={'name':'Train1','lat':17.6132382,'lon': 78.4707318}
47.     pub(mydata)
48.     time.sleep(3)
49.     client.commandCallback=mycommandCallback
50.     client.disconnect()

```

## Output:



The screenshot displays a code editor with Python code on the left and a terminal window on the right. The code defines a configuration, a command callback, and a publish function, then connects to a device and publishes data in a loop. The terminal shows the output of the publish function, displaying the success of each data publication with the device name, latitude, and longitude.

```

File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "i63nvt",
        "devicetypeId": "GPS1",
        "deviceId": "12345"
    },
    "auth": {
        "token": "abodefghij"
    }
}

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=
    print("published data successfully: %s", mydata)

while True:
    mydata={'name':'Train1','lat':11.5892194,'lon':79.3914405}

```

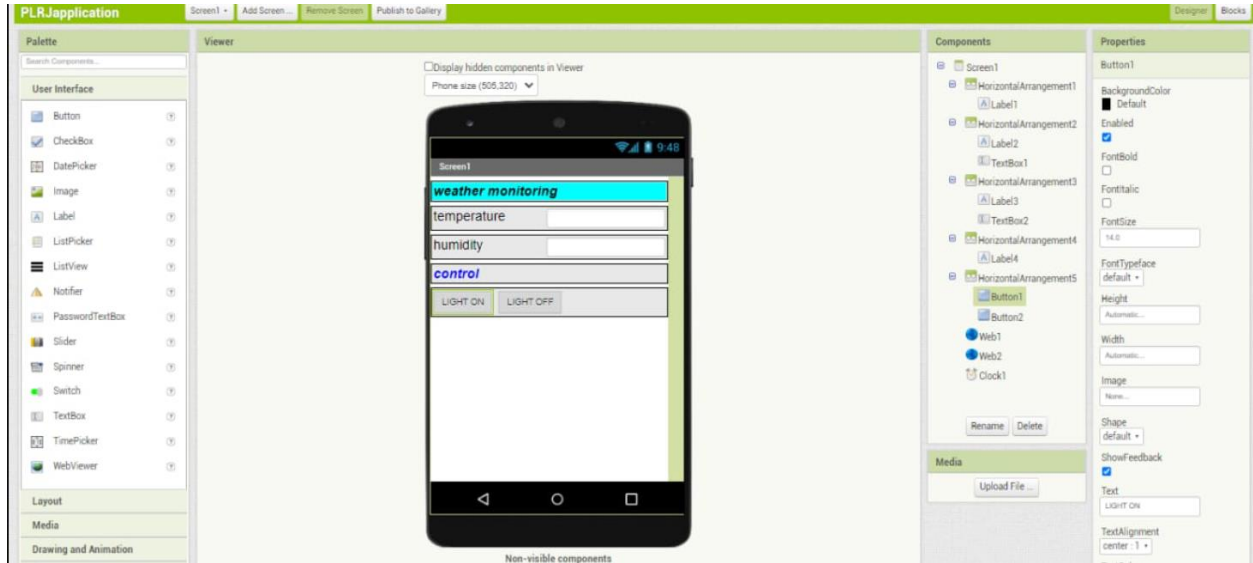
```

File Edit Shell Debug Options Window Help
on': 79.3914405}
published data successfully: %s ('name': 'Train1', 'lat': 11.585909, 'lon': 78.4745052)
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': 11.5892194, 'lon': 79.3914405)
published data successfully: %s ('name': 'Train1', 'lat': 11.585909, 'lon': 11.585909)
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': 11.5892194, 'lon': 79.3914405)
published data successfully: %s ('name': 'Train1', 'lat': 11.585909, 'lon': 11.585909)
published data successfully: %s ('name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052)

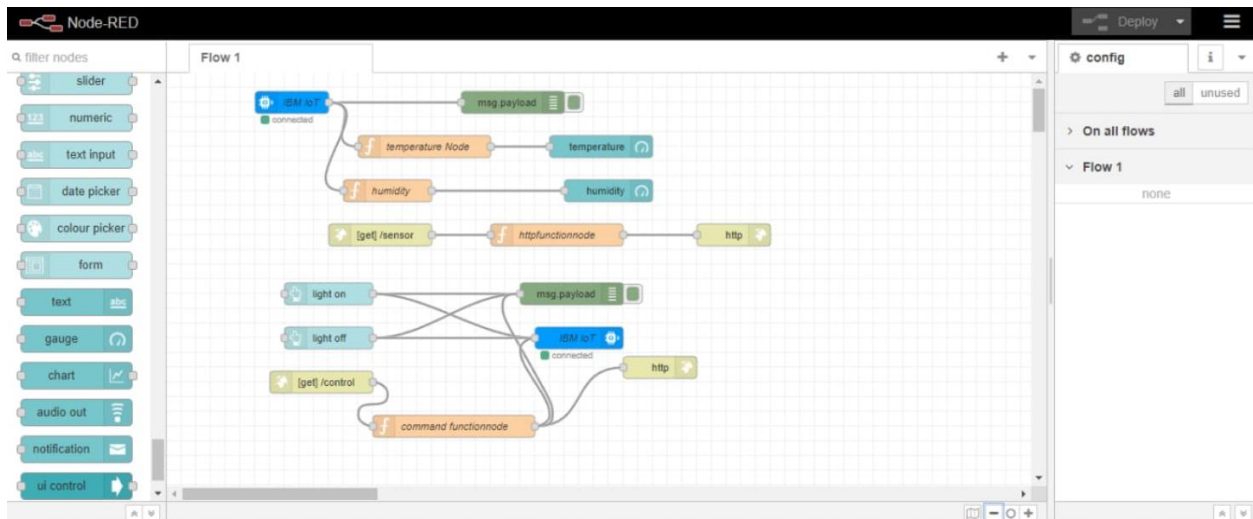
```

## MIT INVERTER:

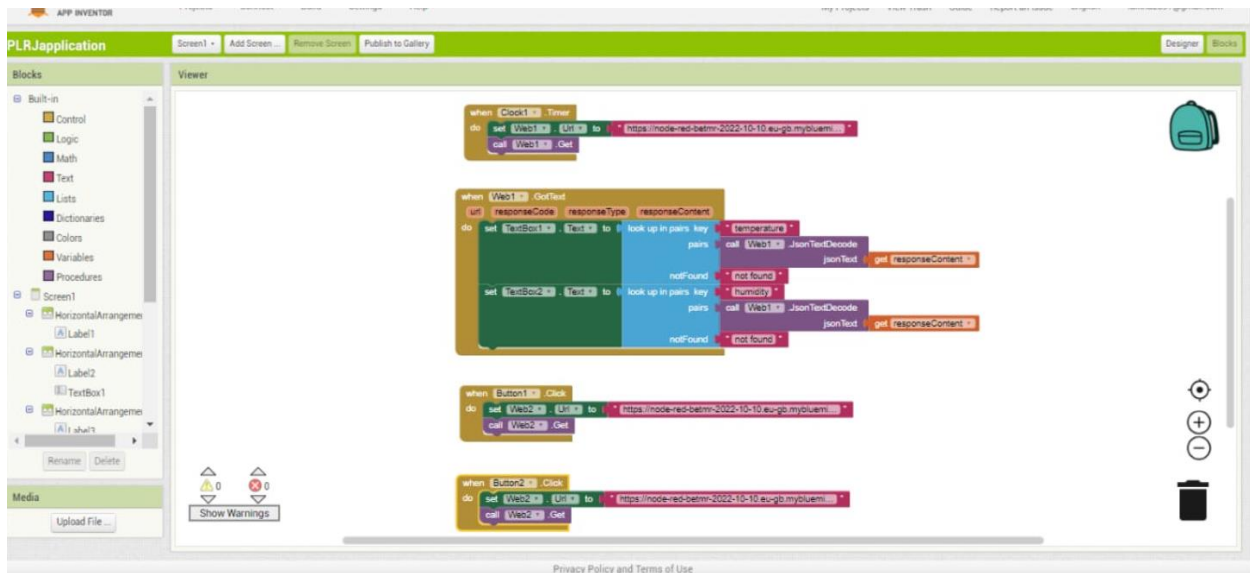
### App design model:



### Node red flow:



### Create mit app inverter:



**Connecting mit app inverter:**



## Connect bar code:



## Software screen mobile phone:

