

SPRINT-4

Date	12 NOVEMBER 2022
Team ID	PNT2022TMID43868
Project Name	Project – SMART SOLUTIONS FOR RAILWAYS

PROCEDURE:

Step1: Develop a node red application for GPS

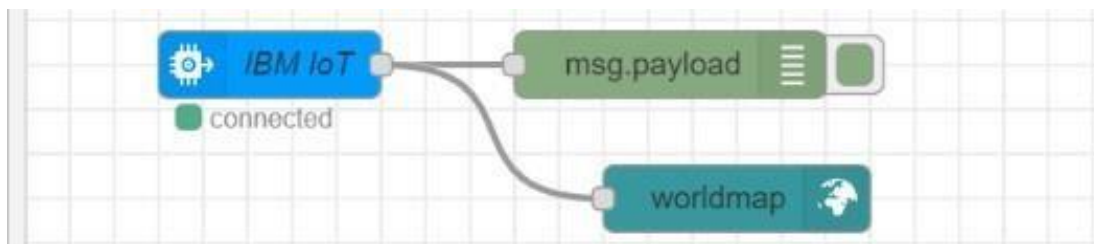
Step2: Develop a python code for GPS

Step3: Run the program

Step4: Train location will be displayed

Step5: Create a node red for wakeup call and E-catering service

NODE RED FLOW:



PYTHON CODE FOR GPS:

```
import wiotp.sdk.device
```

```
import time import random
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId": "sks72l",
```

```
        "typeId": "Sinan",
```

```
        "deviceId": "55"
```

```
    },
```

```
    "auth": {
```

```
        "token": "sinan2000@"
```

```
    }
```

```
}
```

```
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':13.08363 , 'lon': 80.27080}
```

```
    pub (myData)          time.sleep (2)
```

```
    myData={'name': 'Train2', 'lat': 12.40797, 'lon': 79.81410}
```

```
    pub (myData)          time.sleep (2)
```

```
myData={'name': 'Train1', 'lat': 11.83331, 'lon': 79.37465}
```

```
    pub(myData)
```

```
    time.sleep(6)
```

```
    myData={'name': 'Train1', 'lat': 11.59664, 'lon': 78.69899}
```

```
    pub (myData)          time.sleep (6)
```

```
myData={'name': 'Train1', 'lat': 11.63431, 'lon': 78.11122}
```

```
    pub (myData)
```

```
time.sleep (6)
```

```
myData={'name': 'Train1', 'lat':
```

```
11.32207, 'lon': 77.61684}
```

```
pub (myData)          time.sleep (6)
```

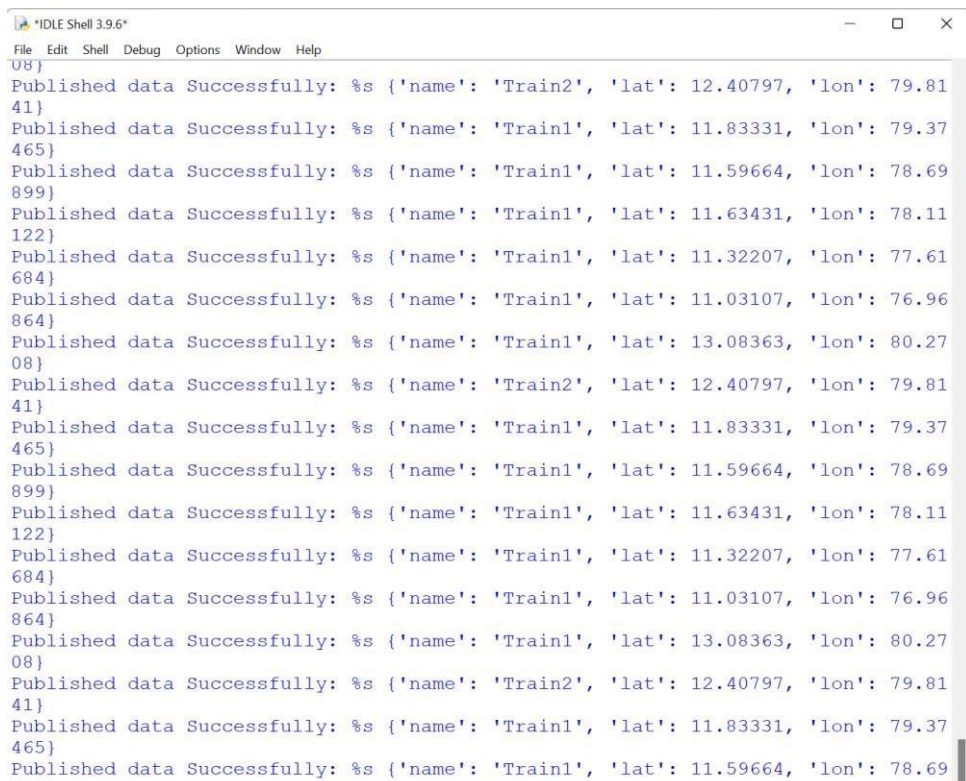
```
myData={'name': 'Train1', 'lat': 11.03107, 'lon': 76.96864}
```

```
pub (myData)  time.sleep (6)          client.commandCallback =
```

```
myCommandCallback
```

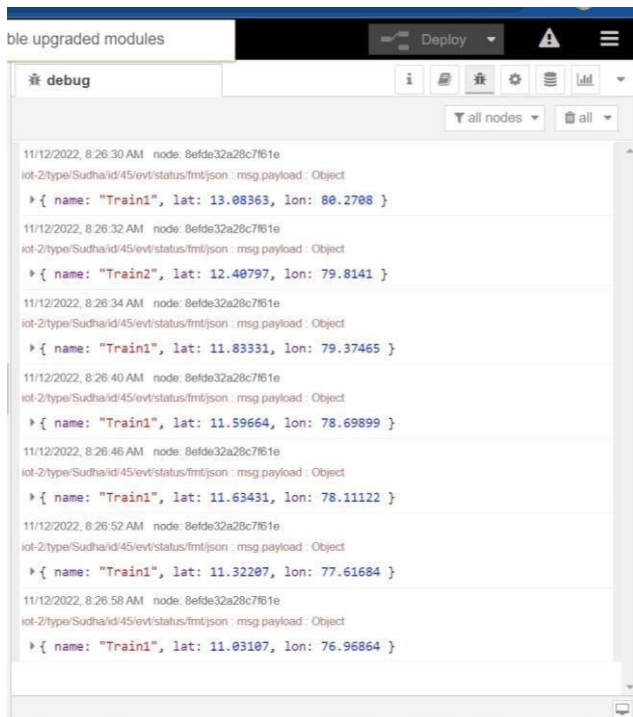
```
client.disconnect ()
```

PYTHON CODE OUTPUT:

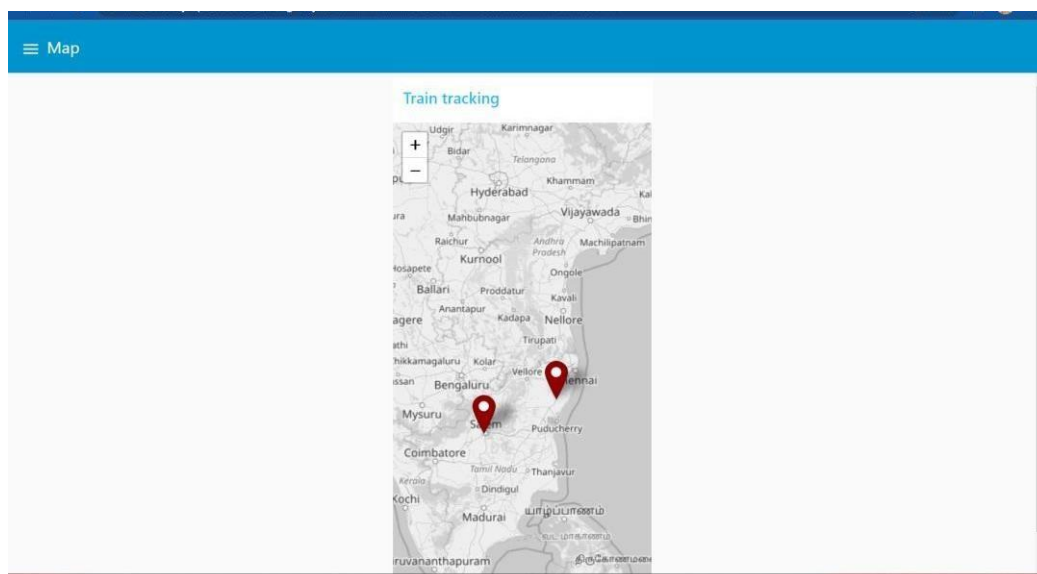
A screenshot of the IDLE Shell 3.9.6 window. The window title is "IDLE Shell 3.9.6". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell shows a series of "Published data Successfully:" messages, each followed by a JSON object containing 'name', 'lat', and 'lon' values. The messages are repeated in a sequence that alternates between 'Train1' and 'Train2' data points. The output is as follows:

```
U8}
Published data Successfully: %s {'name': 'Train2', 'lat': 12.40797, 'lon': 79.81
41}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.83331, 'lon': 79.37
465}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.59664, 'lon': 78.69
899}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.63431, 'lon': 78.11
122}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.32207, 'lon': 77.61
684}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.03107, 'lon': 76.96
864}
Published data Successfully: %s {'name': 'Train1', 'lat': 13.08363, 'lon': 80.27
08}
Published data Successfully: %s {'name': 'Train2', 'lat': 12.40797, 'lon': 79.81
41}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.83331, 'lon': 79.37
465}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.59664, 'lon': 78.69
899}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.63431, 'lon': 78.11
122}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.32207, 'lon': 77.61
684}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.03107, 'lon': 76.96
864}
Published data Successfully: %s {'name': 'Train1', 'lat': 13.08363, 'lon': 80.27
08}
Published data Successfully: %s {'name': 'Train2', 'lat': 12.40797, 'lon': 79.81
41}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.83331, 'lon': 79.37
465}
Published data Successfully: %s {'name': 'Train1', 'lat': 11.59664, 'lon': 78.69
```

NODE RED OUTPUT:



TRAIN TRACKING :



NODE RED CONNECTION FOR WAKEUP CALL AND E-CATERING SERVICE:

≡ wakeup call

wakeup call

Wake up call

☐ wake up call needed

Mobile number *

SUBMIT

CANCEL

≡ E-CATERING

FOOD

Food

VEG

NON-VEG

SUBMIT

CANCEL