# Project Development Phase Sprint-3

| | |
|---|---|
| Date | 11 November 2022 |
| Team ID | PNT2022TMID21782 |
| Project Name | Project: Signs with Smart Connectivity for Better Road Safety. |
| Marks | 20 Marks |

| Sprint | Functional Requirement | User Story Number | User Story/Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-3 | | US-1 | Develop a python script to publish random sensor data such as temperature, humidity, visibility to the IBM IoT platform. | 7 | High | S. Ksheeraja<br>S. Lakshmi Priya<br>B. Menaga<br>V. Ragul |
| Sprint-3 | | US-2 | After developing python code, commands are received print the statements which represent the control of the devices. | 5 | Medium | S. Ksheeraja<br>S. Lakshmi Priya<br>B. Menaga<br>V. Ragul |
| Sprint-3 | | US-3 | Publish Data to the IBM Cloud. | 8 | High | S. Ksheeraja<br>S. Lakshmi Priya<br>B. Menaga<br>V. Ragul |

## US-1 Develop a python script to publish random sensor data such as temperature, humidity and visibility to the IBM IoT Platform

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson DeviceCredentials
organization = "cbp14d"
deviceType = "PNT2022TMID21782"
deviceId = "PNT2022TMID21782"
authMethod = "token"
authToken = "1234567890"


#Intialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s % cmd.data['command']")
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print("led is off")

    #print(cmd)
 try:
    deviceOptions = {"org": organization,"type":
deviceType,"id":deviceId,"authmethod":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #..............................

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
sys.exit()

    # Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times
 deviceCli.connect()
```

```
while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    humid=random.randint(0,100)

    visi=random.randint(0,100)

    data = {'temperature'=temp, 'humidity'=humid,'visibility'=visi}
    #print data
    def myOnPublishCallback():
        print("Published temperature=%s C" %temp,"humidity =%s %%"
%humid,"visibility =%s %%" %visi,"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor","json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
            time.sleep(1)

            deviceCli.commandCallback= myCommandCallback

            #Disconnect the device and application from the cloud
             deviceCli.disconnect()
```