**Project Development Phase**

**Sprint-3**

| Date | 12 November 2022 |
| --- | --- |
| Team ID | PNT2022TMID47460 |
| Project Name | Project - Industry-Specific Intelligent Fire Management System |
| Maximum Marks | 20 Marks |

| **Sprint** | **Functional Requirement (Epic)** | **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, Flame level and Gas level to the IBM IoT platform | 7 | High | IndhumathiHariharan Athira  Arun Raj G |
| Sprint-3 |  | US-2 | After developing python code, commands are received just print the statements which represent the control of the devices. | 5 | Medium | IndhumathiHariharan Athira  Arun Raj G |
| Sprint-3 |  | US-3 | Publish Data To The IBM Cloud | 8 | High | IndhumathiHariharan Athira  Arun Raj G |

**US - 1 Develop a python script to publish random sensor data such as temperature, Flame level and Gas level to the IBM IoT platform**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "4aqwut"

deviceType = "B11M3device\_type"

deviceId = "B11M3device\_id"

authMethod = "token"

authToken = "RcBQ414CD\_p+wKLw+v"

# Initialize GPIO

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

status=cmd.data['command']

if status=="sprinkleron":

print ("Sprinkler is on")

elif status == "sprinkleroff":

print ("Sprinkler is off")

elif status == "exhaustfanon":

print ("Exhaust Fan ON")

elif status == "exhaustfanoff":

print ("Exhaust Fan OFF")

#print(cmd)

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": 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)

flame\_level=random.randint(0,100)

gas\_level = random.randint(0,100)

data = { 'Temperature' : temp, 'Flame\_Level' : flame\_level, 'Gas\_Level' : gas\_level }

#print data

def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Flame\_Level = %s %%" % flame\_level, "Gas\_Level = %s %%" %gas\_level ,"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()

**.**

2022-11-08 07:26:20,139 ibmiotf.device.Client INFO Connected successfully: d:4aqwut:B11M3EDEVICETYPE:B11M3DEVICEID