

PROJECT DEVELOPMENT PHASE

SPRINT-3

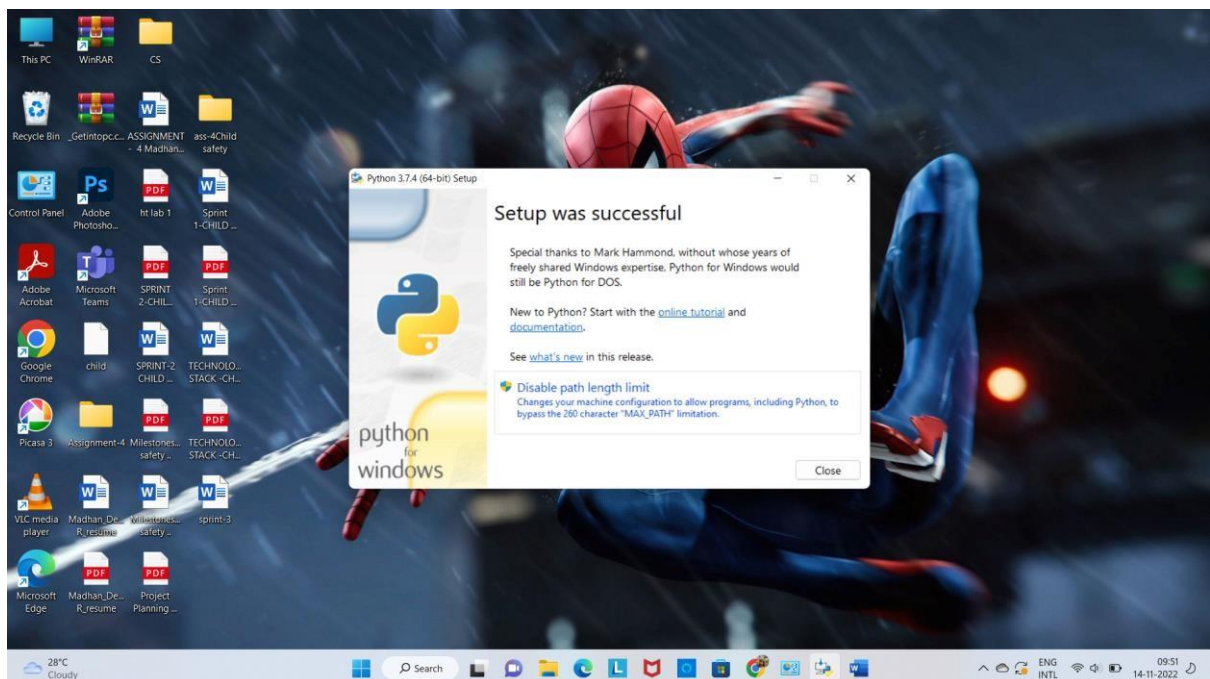
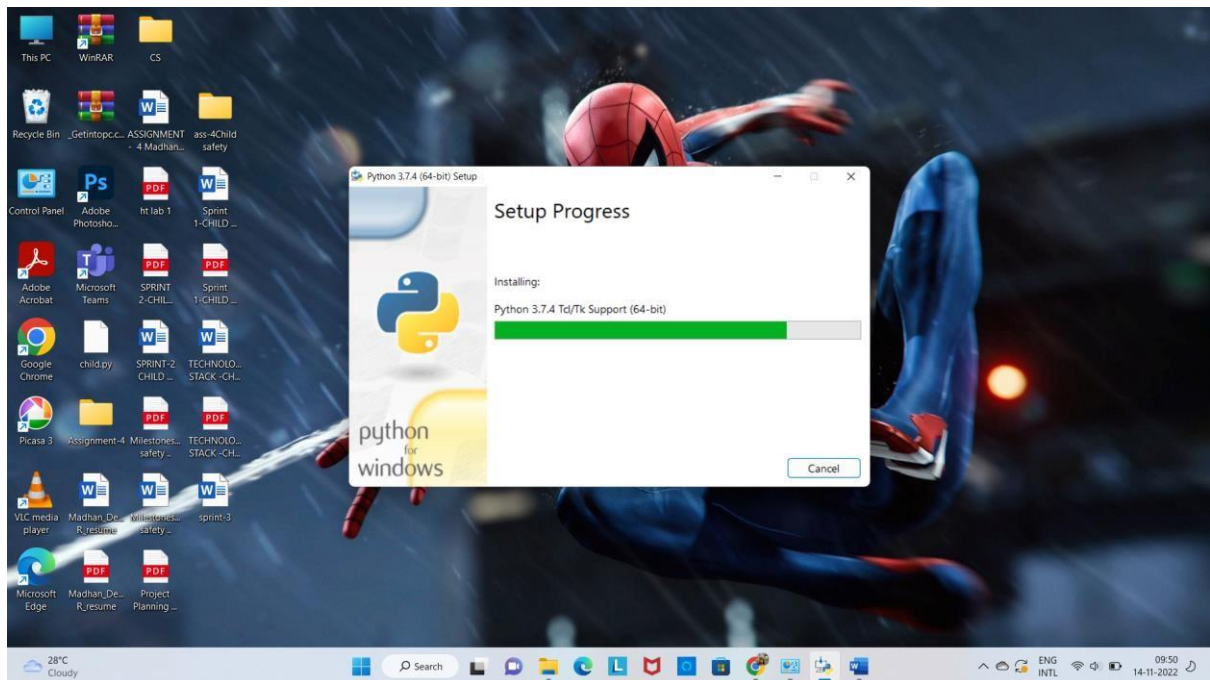
Date	12 NOVEMBER 2022
Team ID	PNT2022TMID46169
Project Name	IOT BASED SAFETY GADGET FOR CHILD MONITORING AND NOTIFICATION

Launch the Cloudant DB and Create database to store the location data:

The screenshot shows the IBM Cloud console interface. The top navigation bar includes 'Resource list', 'Search resources and products...', 'Catalog', 'Manage', and the user name 'MADHAN DEEPAK R's...'. The main content area displays the details for a resource named 'node-red-rxzvj-2022--cloudant-1666687445120'. The 'Overview' tab is selected, showing deployment details such as CRN, Location (London), External endpoint, and Authentication methods. A 'Launch Dashboard' button is visible in the top right corner of the resource details section.

The screenshot shows the Cloudant dashboard interface. The top navigation bar includes 'Databases', 'Database name', 'Create Database', and 'JSON'. The main content area displays a table of databases. The table has columns for Name, Size, # of Docs, Partitioned, and Actions. A single database named 'nodered' is listed with a size of 22 bytes and 1 document. The 'Actions' column for this database contains icons for edit, lock, and delete. The bottom of the dashboard shows 'Showing 1-1 of 1 databases' and 'Databases per page 20'.

Install the python software:



Develop the python scripts to publish details to IBM IoT Platform

```
child.py - C:/Python/Python37/child.py (3.7.4)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "0rfzob"
deviceType = "NodeREDMCU"
deviceId = "NodeRed123456"
authMethod = "token"
authToken = "123456789"

# Initialize 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, "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:
```

```
child.py - C:/Python/Python37/child.py (3.7.4)
File Edit Format Run Options Window Help

        print ("led is on")
    else :
        print ("led is 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
    latitude=random.randint(0,100)
    longitude=random.randint(0,100)

    data = { 'latitude' : latitude, 'longitude': longitude }
    #print data
    def myOnPublishCallback():
        print ("Published latitude = %s " % latitude, "longitude = %s " % longitude,"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

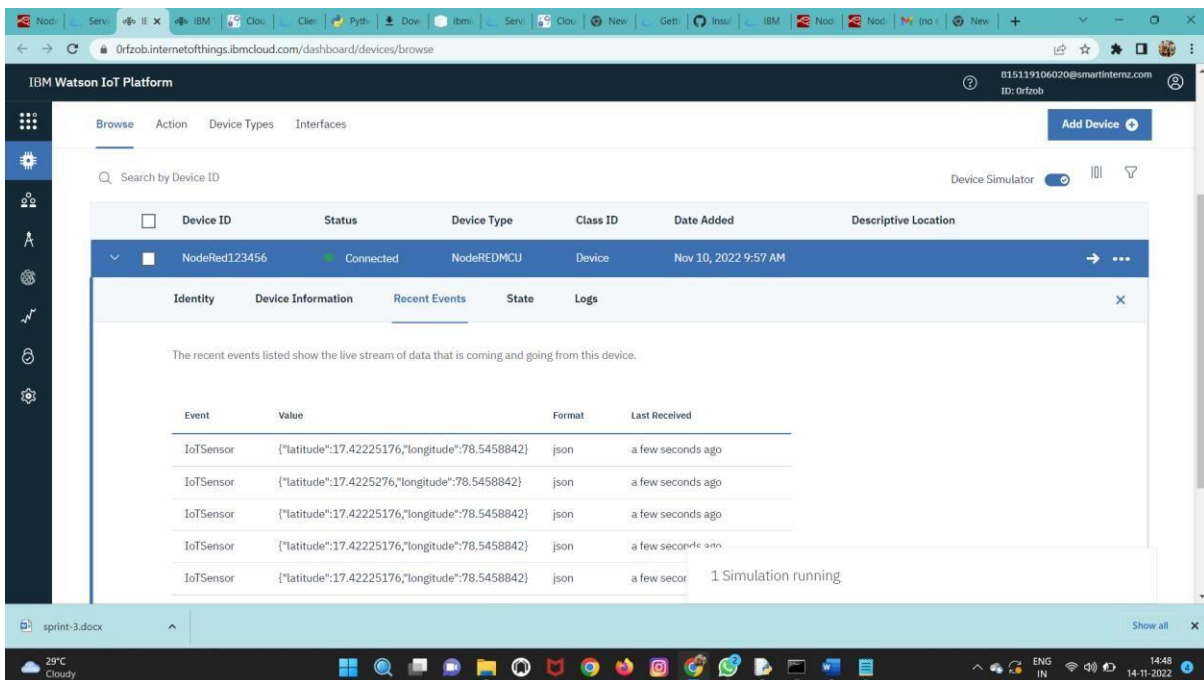
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Integrate the device id, authentication token in python script:

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

#Provide your IBM Watson Device Credentials
organization = "0rfzob"
deviceType = "NodeREDMCU"
deviceId = "NodeRed123456"
authMethod = "token"
authToken = "123456789"
```

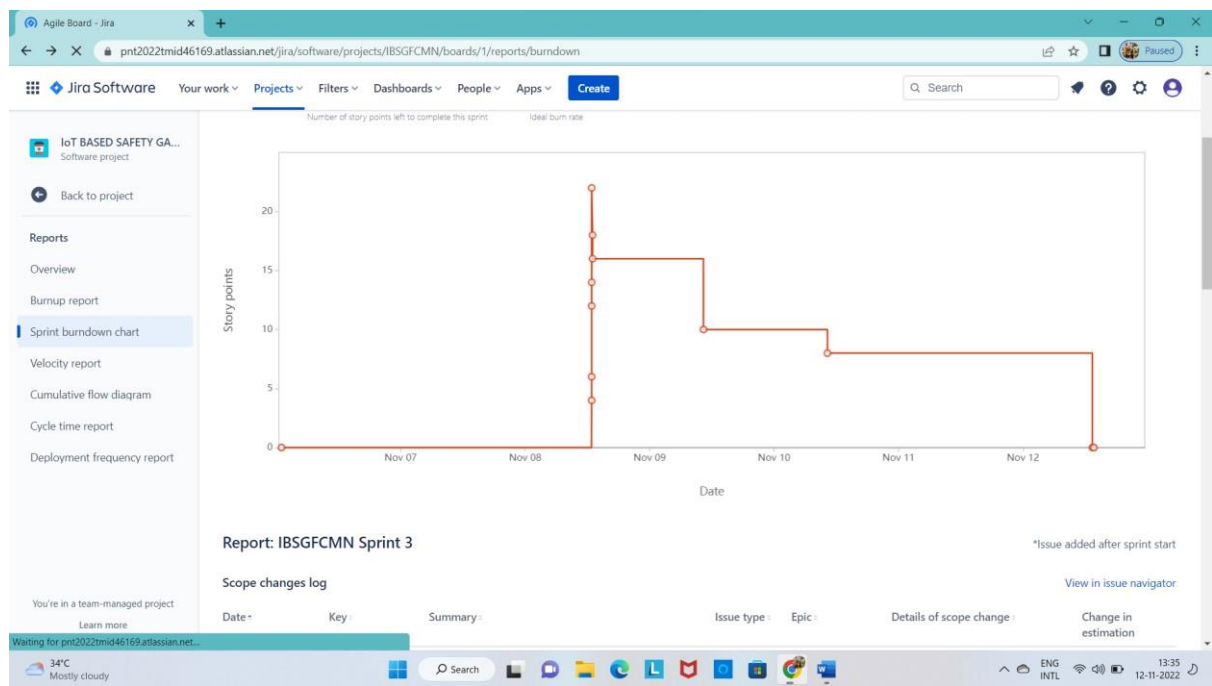
Develop the python code for publishing the location (latitude & longitude) to IBM IoT Platform:



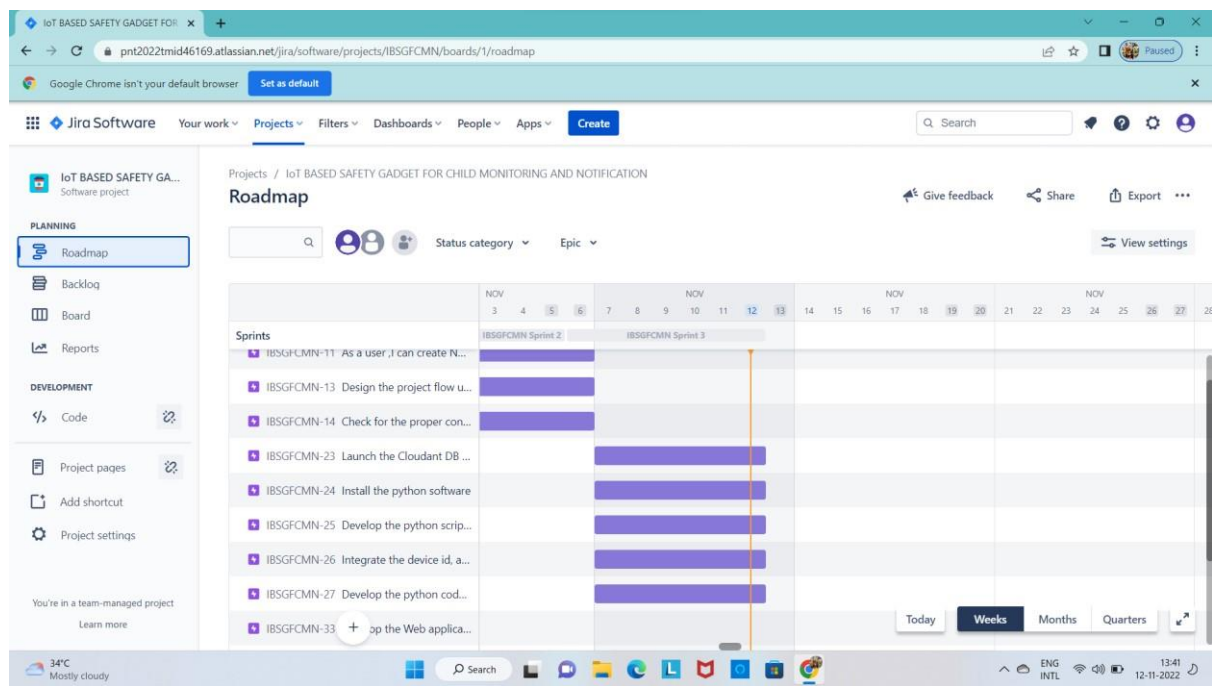
The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area shows a table of devices. The first device, 'NodeRed123456', is highlighted and its details are expanded. The 'Recent Events' tab is selected, showing a list of events. Each event contains a JSON object with 'latitude' and 'longitude' values. The 'Last Received' column indicates the time since the event was received. A status message at the bottom right of the events table states '1 Simulation running'.

Event	Value	Format	Last Received
IoTSensor	{"latitude":17.42225176,"longitude":78.5458842}	json	a few seconds ago
IoTSensor	{"latitude":17.4225276,"longitude":78.5458842}	json	a few seconds ago
IoTSensor	{"latitude":17.42225176,"longitude":78.5458842}	json	a few seconds ago
IoTSensor	{"latitude":17.42225176,"longitude":78.5458842}	json	a few seconds ago
IoTSensor	{"latitude":17.42225176,"longitude":78.5458842}	json	a few seconds ago

BURNDOWN CHART:



ROAD MAP:



VELOCITY REPORT:

