

Team ID	PNT2022TMID35389
Project Name	Smart waste management system for metropolitan cities

## Sprint 1:

### DEVELOPMENT OF PYTHON CODE:

```
import time
import sys
import ibmiotf.device
import random
from ibmcloudant.cloudant_v1 import Document, CloudantV1, BulkDocs
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('YainZLYNqB_hRLBdq-1xI_nX0h3RhaUsxXgmOSav6Yof')

service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://9f90bf29-84c3-49c2-b540-44822170d56b-
bluemix.cloudantnosqldb.appdomain.cloud/')

organisation="usx5i2"
deviceType="dustbin"
deviceId="1"
authMethod="token"
authToken="123456789"

# a-js87ki-4h8q4ploiq
# GiQhbtWSUHg1E2g3K0
dustbin_locations=[[13.012594155082645,80.23527327140268],[13.01077265635586,
80.23634347276455],[13.012047968197532, 80.23251864275869],[13.013349406731104,
80.2334520515062]]
def get_bin_value_weight():
    final_value_and_weight=[]
    bin_value_list=[]
    bin_weight_list=[]
    for i in range(100):
        bin_value_list.append(random.randint(0,100))
        bin_weight_list.append(random.randint(0,25))
    bin_value_list.sort()
    bin_weight_list.sort()
    for i in range(100):
        final_value_and_weight.append([bin_value_list[i],bin_weight_list[i]])
    return final_value_and_weight
```

```

try:
    deviceOptions={"org":organisation,"type":deviceType,"id":deviceId,"auth-
method":authMethod,"auth-token":authToken}
    deviceCli=ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device!")
    sys.exit()
deviceCli.connect()

#Get initial rev value
response = service.post_all_docs(
    db='dustin_value',
    key='dustbin_val',
    limit=10
).get_result()
rev=response['rows'][0]['value']['rev']

while True:
    all_sensors_values=[]
    for i in range(len(dustbin_locations)):
        bin_value_weight=get_bin_value_weight()
        all_sensors_values.append(bin_value_weight)
    for i in range(len(all_sensors_values[0])):
        bin_value_1=all_sensors_values[0][i][0]
        bin_weight_1=all_sensors_values[0][i][1]
        bin_value_2=all_sensors_values[1][i][0]
        bin_weight_2=all_sensors_values[1][i][1]
        bin_value_3=all_sensors_values[2][i][0]
        bin_weight_3=all_sensors_values[2][i][1]
        bin_value_4=all_sensors_values[3][i][0]
        bin_weight_4=all_sensors_values[3][i][1]

    data={'lat_1':dustbin_locations[0][0],'long_1':dustbin_locations[0][1],
        'lat_2':dustbin_locations[1][0],'long_2':dustbin_locations[1][1],
        'lat_3':dustbin_locations[2][0],'long_3':dustbin_locations[2][1],
        'lat_4':dustbin_locations[3][0],'long_4':dustbin_locations[3][1],
        'bin_value_1':bin_value_1,'bin_weight_1':bin_weight_1,
        'bin_value_2':bin_value_2,'bin_weight_2':bin_weight_2,
        'bin_value_3':bin_value_3,'bin_weight_3':bin_weight_3,
        'bin_value_4':bin_value_4,'bin_weight_4':bin_weight_4
    }
    products_doc = Document(
        id="dustbin_val",
        rev=rev,
        lat_1=dustbin_locations[0][0],long_1=dustbin_locations[0][1],
        lat_2=dustbin_locations[1][0], long_2=dustbin_locations[1][1],
        lat_3=dustbin_locations[2][0], long_3=dustbin_locations[2][1],
        lat_4=dustbin_locations[3][0], long_4=dustbin_locations[3][1],
        bin_value_1=bin_value_1, bin_weight_1= bin_weight_1,

```

```
bin_value_2= bin_value_2, bin_weight_2= bin_weight_2,
bin_value_3= bin_value_3, bin_weight_3=bin_weight_3,
bin_value_4= bin_value_4, bin_weight_4= bin_weight_4)
def myOnPublishCallback():
    print(data)
    pass
    success=deviceCli.publishEvent("IoTSensor","json",data,qos=0,on_publish=myOnPublishCallback())
    response = service.post_document(db='dustbin_value',
document=products_doc).get_result()
    rev = response['rev']
    if not success:
        print("Failed!!")
    time.sleep(1)
deviceCli.disconnect()
```

