

# Project Development Phase

## SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

TEAM ID: PNT2022TMID39310

### TEAM MEMBERS

ROLE	TEAM MEMBERS NAME	ROLL NO
TEAM LEADER	RUBESH .S	(422619104035)
TEAM MEMBER 1	ABIRAMI .S	(422619104002)
TEAM MEMBER 2	HARITHA .S	(422619104016)
TEAM MEMBER 3	TAMIZHSELVAN .D	(422619104044)

## Project Development - Delivery Of Sprint-3

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-3	Develop A Python Script	USN- 6	Develop A Python Script	10	High	S.Rubesh S.Haritha S.Abirami D.Tamizhselvan
Sprint-3	Develop A Python Script	USN- 7	Publish Data To The IBM Cloud	10	Medium	S.Rubesh S.Haritha S.Abirami D.Tamizhselvan

## Delivery

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022

# Develop a python code for publishing the location (latitude and longitude) data along with bin values to the IBM IoT Platform.

**Task Assigned:** S.Rubesh,S.Haritha,S.Abirami,D.Tamizhselvan

**Task Started On:** 07 Nov 2022

**Task Completion Date:** 09 Nov 2022

## Installation:

Use Anaconda Prompt command prompt/terminal from your machine to ensure you can import geocoders successfully

### **pip install geopy**

```
Requirement already satisfied: geopy in
/Users/TS/opt/anaconda3/lib/python3.7/site-packages (2.0.0)
Requirement already satisfied: geographiclib<2,>=1.49 in
/Users/TS/opt/anaconda3/lib/python3.7/site-packages (from geopy) (1.50)
Note: you may need to restart the kernel to use updated packages.
```

### **#prereq libraries**

```
import pandas as pd
from geopy.geocoders import Nominatim
#read-in csv file and create column headers according to UK housing site
df = pd.read_csv("data.csv", #file name
                sep=',') #seperator
# print the data
df.head()
```

```
#create variable column inside file called `myAddress` and assign it to
selected address columns
```

```
df['query'] = df['COMPANY'] + " " + df['CITY'] + " " + df['COUNTRY']
```

```
#print new column with index -first 5 rows only
df.iloc[0:5, 8:9]
```

```
#remove duplicate addresses (new concat column) but keep first instance
df.drop_duplicates(subset='query', keep='first', inplace=True)
```

```
#print some useful info: row length and shape size
print("data row x columns is {} \n data row count is {}".format(df.shape, len(df.index)))
```

```
#print first rows as sample
df.head()
```

```

#drop columns we won't use
df = df.drop(columns=['DISTRICT', 'COUNTY'])

#print row length and shape size
print("data row x columns is {} \ndata row count is {}".format(df.shape, len(df.index)))

#print first rows as sample
df.head()

#create 2 new columns to store lat/long - initialise to null
df['location_lat'] = ""
df['location_long'] = ""
df['location_address'] = ""

#print first rows to sample
df.head()

'''
**Get Lat/Long Data with GeoPy**
-----

the code below calls a geopy API using a concatenated column of address
values. We use this column as a query key
to pull back cooresponding lat/long coordinates.
'''

geolocator = Nominatim(user_agent="myApp")

for i in df.index:
    try:
        #tries fetch address from geopy
        location = geolocator.geocode(df['query'][i])

        #append lat/long to column using dataframe location
        df.loc[i, 'location_lat'] = location.latitude
        df.loc[i, 'location_long'] = location.longitude
        df.loc[i, 'location_address'] = location.address
    except:
        #catches exception for the case where no value is returned
        #appends null value to columns
        df.loc[i, 'location_lat'] = ""
        df.loc[i, 'location_long'] = ""
        df.loc[i, 'location_address'] = ""

#print first rows as sample
df.head()

#write the contents thus far to new csv file
df.to_csv('geopy_data.csv')

```