|  |
| --- |
| **ASSIGNMENT-2** |
| ASSIGNMENT **: Build Python code, Generate Temperature and Humidity values (Use Random function to generate values) and**  **write a condition to detect an alarm in case of high temperature and high Humidity** |
| NAME:K.PRABHU |

PROGARM:

#1 import try:

import configparser except:

from six.moves import configparser

import smtplib

from email.mime.multipart import MIMEMultipart from email.mime.text import MIMEText

import requests

#2 variable related to weather API

weather\_dict = {'freezing\_rain\_heavy': 'Heavy rain and snow', 'freezing\_rain': 'Rain and snow', 'freezing\_rain\_light': 'Light rain and snow', 'freezing\_drizzle': 'Light drizzle and snow', 'ice\_pellets\_heavy': 'Heavy ice pellets', 'ice\_pellets': 'Normal ice pellets', 'ice\_pellets\_light': 'Light ice pellets', 'snow\_heavy': 'Heavy snow', 'snow': 'Normal snow', 'snow\_light': 'Light snow', 'tstorm': 'Thunder storm', 'rain\_heavy': 'Heavy rain', 'rain': 'Normal rain', 'rain\_light': 'Light rain'}

url = "https://api.climacell.co/v3/weather/nowcast"

querystring =

{"lat":"1.29027","lon":"103.851959","unit\_system":"si","timestep":"60","start\_time":"now","fields":"te mp,humidity,weather\_code","apikey":"xxxx"}

#3 class

class EmailSender(): #4 initialization def init (self):

self.cf = configparser.ConfigParser() self.cf.read('./config.ini')

self.sec = 'email'

self.email = self.cf.get(self.sec, 'email') self.host = self.cf.get(self.sec, 'host') self.port = self.cf.get(self.sec, 'port')

self.password = self.cf.get(self.sec, 'password')

#5 main function to send email def SendEmail(self, recipient): title = "Home Sweet Home"

#6 create a new multipart mime object msg = MIMEMultipart()

msg['Subject'] = '[Weather Notification]'

msg['From'] = self.email msg['To'] = ', '.join(recipient)

#7 call weather API using requests

response = requests.request("GET", url, params=querystring) result = ""

json\_data = response.json() #print(json\_data)

#8 loop over each data and check for abnormal weather (rain, snow) for i in range(len(json\_data)):

if(json\_data[i]['weather\_code']['value'] in weather\_dict):

if(i == 0):

result = "%s at the moment. Current temperature is " % (weather\_dict[json\_data[i]['weather\_code']['value']])

else:

result = "%s in %s hour(s) time. Forecasted temperature is " % (weather\_dict[json\_data[i]['weather\_code']['value']], i)

result += '%s%s while the humidity is about %s%s' % (json\_data[i]['temp']['value'], json\_data[i]['temp']['units'], json\_data[i]['humidity']['value'], json\_data[i]['humidity']['units'])

msgText = MIMEText('<b>%s</b><p>%s</p>' % (title, result), 'html') msg.attach(msgText)

#9 authenticate and send email

with smtplib.SMTP(self.host, self.port) as smtpObj: smtpObj.ehlo()

smtpObj.starttls() smtpObj.login(self.email, self.password)

smtpObj.sendmail(self.email, recipient, msg.as\_string()) return "Success"

return "Failed" break