Flood Monitoring and Early Warning

System

**code in Python IDE**

* For writing python code we will be using python IDE.
* In this project we will be making two python files. One will be saved in the name of conf.py and other will be main.py.
* The purpose of making two files is to make the code understandable. Also this both python files will be usefull in sending sms and emails alerts to users.
* Now the most important part is arrived writing code in Python IDE. The full code is divided into two parts. The detailed code is given below.
* Open Python 3.7 IDE(Downloaded from the above section).
* Click on new file. Save the file in the name conf.py.
* **conf.py:** The file consists of important Api keys, Device id of Bolt IoT WiFi Module. Also it consists of important keys of Twillo and Mailgun respectively which will be further usefull in this project.
* Below is the complete structure of conf.py file. Make sure that you add the updated Bolt API key, device id and Mailgun and Twillo details respectively:

#twillo details for sending alert sms

SID = 'You can find SID in your Twilio Dashboard'

AUTH\_TOKEN = 'You can find on your Twilio Dashboard'

FROM\_NUMBER = 'This is the no. generated by Twilio. You can find this on your Twilio Dashboard'

TO\_NUMBER = 'This is your number. Make sure you are adding +91 in beginning'

#bolt iot details

API\_KEY = 'XXXXXXXXX'

#This is your Bolt cloud API

Key.

DEVICE\_ID = 'BOLTXXXXXXXXX' #This is the ID of your Bolt device.

#mailgun details for sending alert E-mails

MAILGUN\_API\_KEY = 'This is the private API key which you can find on your Mailgun Dashboard'

SANDBOX\_URL= 'You can find this on your Mailgun Dashboard'

SENDER\_EMAIL = 'test@ + SANDBOX\_URL' # No need to modify this. The sandbox URL is of the format test@YOUR\_SANDBOX\_URL

RECIPIENT\_EMAIL = 'Enter your Email ID Here'

* After writing the conf.py now the last part is to write the main.py code. This code will be helpfull to send sms and email alerts when the water level crosses the threshold.
* Open the Python IDE.
* Click on new file. Save the file in the name main.py. Save the file in the same path where conf.py is saved.
* **main.py:** This file consists of the main coding facility. Discussed earlier it will be used to send sms and emails alerts. It will be also helpfull to keep close monitor on water level to send alerts whenever required.
* Below is the complete code of main.py.

import conf  
from boltiot import Sms, Email, Bolt  
import json, time  
  
intermediate\_value = 55  
max\_value = 80  
  
  
mybolt = Bolt(conf.API\_KEY, conf.DEVICE\_ID)  
sms = Sms(conf.SID, conf.AUTH\_TOKEN, conf.TO\_NUMBER, conf.FROM\_NUMBER)  
mailer = Email(conf.MAILGUN\_API\_KEY, conf.SANDBOX\_URL, conf.SENDER\_EMAIL, conf.RECIPIENT\_EMAIL)  
  
  
def twillo\_message(message):  
 try:  
 print("Making request to Twilio to send a SMS")  
 response = sms.send\_sms(message)  
 print("Response received from Twilio is: " + str(response))  
 print("Status of SMS at Twilio is :" + str(response.status))  
 except Exception as e:  
 print("Below are the details")  
 print(e)  
  
def mailgun\_message(head,message\_1):  
 try:  
 print("Making request to Mailgun to send an email")  
 response = mailer.send\_email(head,message\_1)  
 print("Response received from Mailgun is: " + response.text)  
 except Exception as e:  
 print("Below are the details")  
 print(e)  
   
while True:  
 print ("Reading Water-Level Value")  
 response\_1 = mybolt.serialRead('10')  
 response = mybolt.analogRead('A0')  
 data\_1 = json.loads(response\_1)  
 data = json.loads(response)   
 Water\_level = data\_1['value'].rstrip()  
 print("Water Level value is: " + str(Water\_level) + "%")  
 sensor\_value = int(data['value'])  
 temp = (100\*sensor\_value)/1024  
 temp\_value = round(temp,2)  
 print("Temperature is: " + str(temp\_value) + "°C")

try:   
   
 if int(Water\_level) >= intermediate\_value:  
 message ="Orange Alert!. Water level is increased by " +str(Water\_level) + "% at your place. Please be Safe. The current Temperature is " + str(temp\_value) + "°C."  
 head="Orange Alert"  
 message\_1="Water level is increased by " + str(Water\_level) + "% at your place. Please be Safe. The current Temperature is " + str(temp\_value) + "°C."  
 twillo\_message(message)  
 mailgun\_message(head,message\_1)  
  
 if int(Water\_level) >= max\_value:  
 message ="Red Alert!. Water level is increased by " + str(Water\_level) + "% at your place. Please Don't move out of the house. The Current Temperature is " + str(temp\_value) + "°C"  
 head="Red Alert!"  
 message\_1="Water level is increased by " + str(Water\_level) + "% at your place. Please Don't move out of the house. The Current Temperature is " + str(temp\_value) + "°C."  
 twillo\_message(message)  
 mailgun\_message(head,message\_1)  
  
 except Exception as e:   
 print ("Error occured: Below are the details")  
 print (e)  
 time.sleep(15)



