Assignment4 – Temperature Sensor Readings logged to a cloud based Server.

```
#libraries required
import mraa
import argparse
import sqlite3
import requests
import smtplib
import time
import datetime

from time import strftime
from email.mime.text import MIMEText

#declaring format for date time stamp
fmt = "%H:%M:%S"
fmtday = "%Y/%m/%d"

#initialising gpio hardware pins
led_red = mraa.Gpio(2)
led_green = mraa.Gpio(4)
temp_read = mraa.Aio(0)
temp_read.setBit(12)

#directing gpio as output pins
led_red.dir(mraa.DIR_OUT)
led_green.dir(mraa.DIR_OUT)

#initialising sender, receiver, subject and body for email
sender = 'arundhathirs1993@gmail.com'
receiver = 'arundhathirs1993@gmail.com'
sub = 'Temperature Sensor Alert'
body = 'Temperature exceeded Preset Range'
msg = """From: %s \nTo: %s \nSubject: %s\n\n%s """ % (sender, ",".join(receiver), sub, body)

#login credentials
h = "36274aa4a30dbf60ad4bd3f724517003"
e = "arundhathi.swami@colorado.edu"

#connecting to database on server
connection = sqlite3.connect("A4_TempSensor.db")
t = connection.cursor()
```

```
#parsing arguments to database
parser = argparse.ArgumentParser()
parser.add_argument("--upper_limit", help = "lower boundary on acceptable temperature range", type=float)
parser.add_argument("--lower_limit", help = "upper boundary on acceptable temperature range", type=float)
parser.add_argument("--count", help = "no of temp readings taken", default = 0, type = int)

args = parser.parse_args()
t.execute("CREATE TABLE IF NOT EXISTS temp_entries (datetimestamp TEXT, C_deg REAL, upper_limit REAL, lower_limit
REAL)")

counter = (args.count)
print ("\n")
while(1):
   rts = time.time()
   datetimestamp = str(datetime.datetime.fromtimestamp(rts).strftime('%Y/%m/%d %H:%M:%S'))
   read_temp_value  = temp_read.read()
   v = float(read_temp_value/819.0)
   C_deg = ((v*100)-50)
   F_deg = (((C_deg*9.0)/5.0)+32.00)

   url = "https://192.168.1.50/temperature/"
   url = url + "insert.php?h=" +h
   url = url + "&e=" +e
   url = url + "&d=" +str(C_deg)
   requests.post(url,verify = False)

   lower_limit = float(args.lower_limit)
   upper_limit = float(args.upper_limit)

   query="INSERT INTO temp_entries VALUES (?,?,?,?)"
   c=(datetimestamp,C_deg,lower_limit, upper_limit)
   t.execute(query,c)
   connection.commit()
   time.sleep(2)

   if(counter>0):
      counter -= 1
      print ('Temperature in C:', +C_deg)

   if(C_deg < lower_limit or C_deg > upper_limit):
      led_red.write(1)
      led_green.write(0)
      server = smtplib.SMTP('smtp.gmail.com:587')
```

```
        server.ehlo()
        server.starttls()
        server.ehlo()
        server.login("arundhathirs1993@gmail.com", "Rajadibharu23671518")
        server.sendmail(sender,receiver,msg)
        server.quit()
        print ("Temperature Exceeded Range\n")
    else:
        led_green.write(1)
        led_red.write(0)
        print ("Temperature within preset range.\n")

    if (counter ==0):
        avg = "SELECT AVG(C_deg) AS average FROM temp_entries"
        max_temp = "SELECT MAX(C_deg) FROM temp_entries"
        min_temp = "SELECT MIN(C_deg) FROM temp_entries"
        t.execute(avg)
        avg = t.fetchall()
        t.execute(max_temp)
        max_temp = t.fetchall()
        t.execute(min_temp)
        min_temp = t.fetchall()

        print"Avg Temperature Recorded is: {} \n Max Temperature Recorded is: {} \n Min Temperature Recorded is:
{}".format(avg,max_temp,min_temp)

        counter = args.count
```

/_____/

Mail Sent: