TEMPERATURE DETECTORS

Problem Statement:

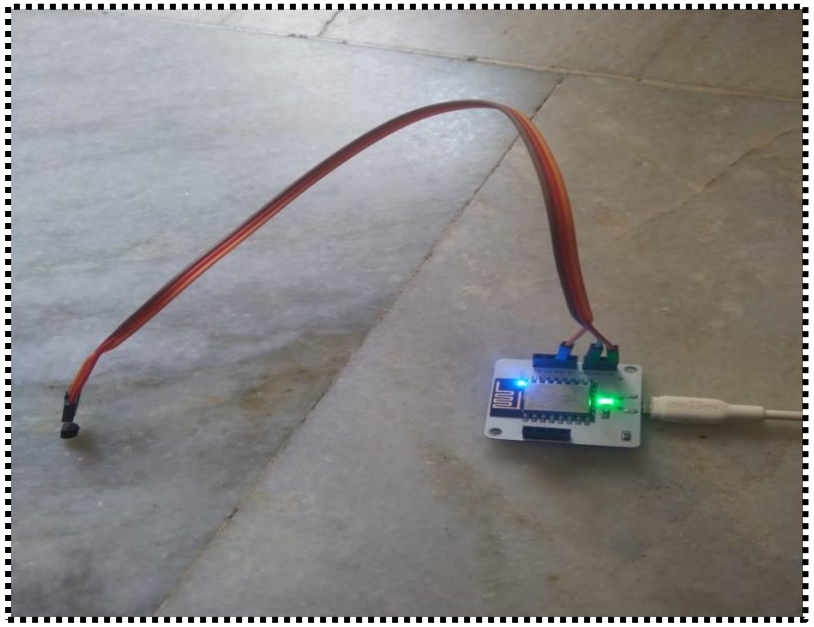
In Pharmaceutical companies, they maintain the temperatures of the capsules in the range of threshold values. If theis threshold values crosses their limits, then there will be a huge loss to the companies.

So, to minimize this loss, we come up with a solution by using IOT.

Solution:

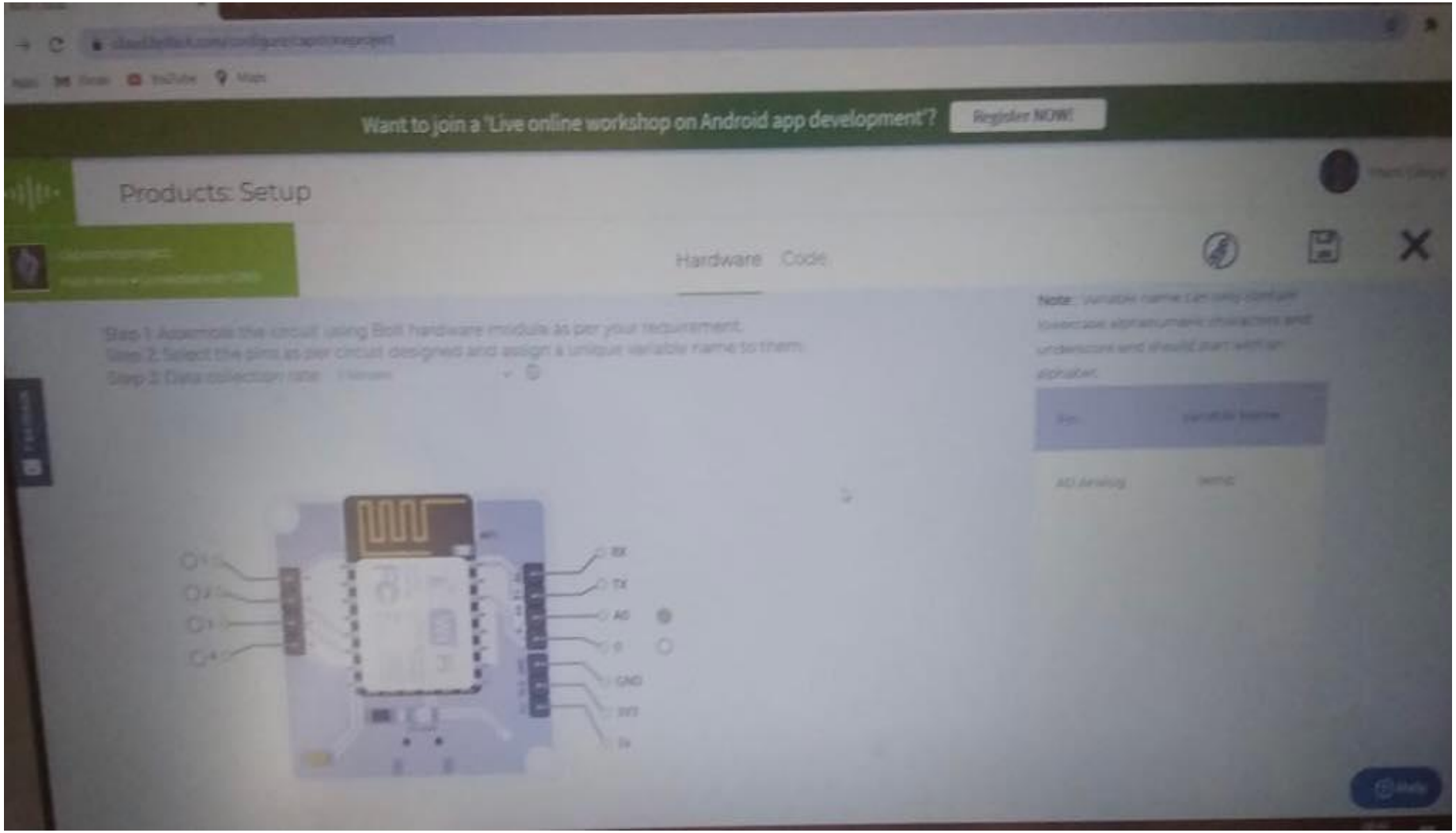
I have designed a solution in such a way that, whenever the temperature crosses the range of threshold values, the company owner will receive an alert of warning message either through SMS or through Email. So that He/ She can immediately set the temperature back to its threshold limits. By this, the loss will be decreases.

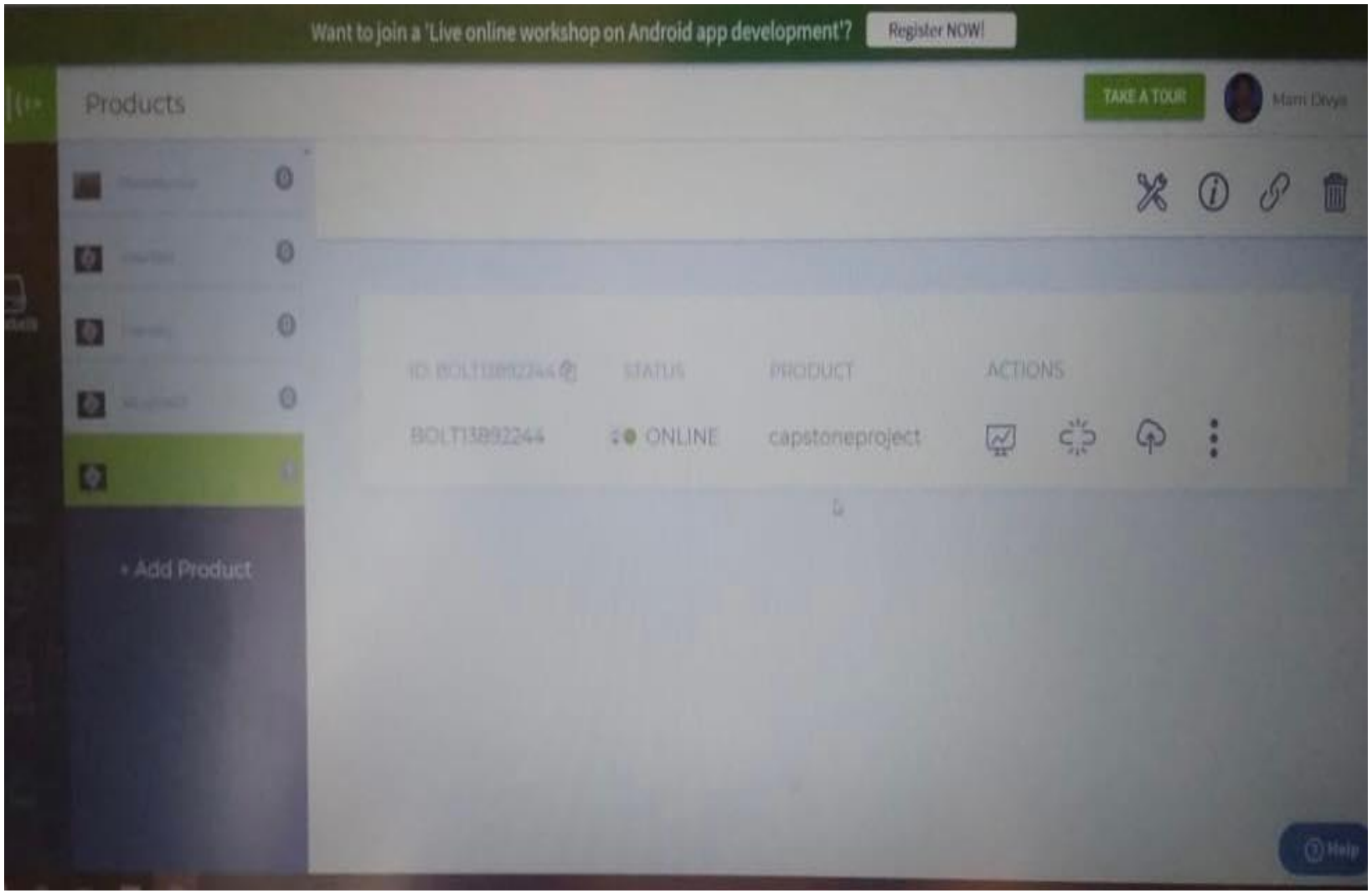
Here are some of the shots that taken while doing the project



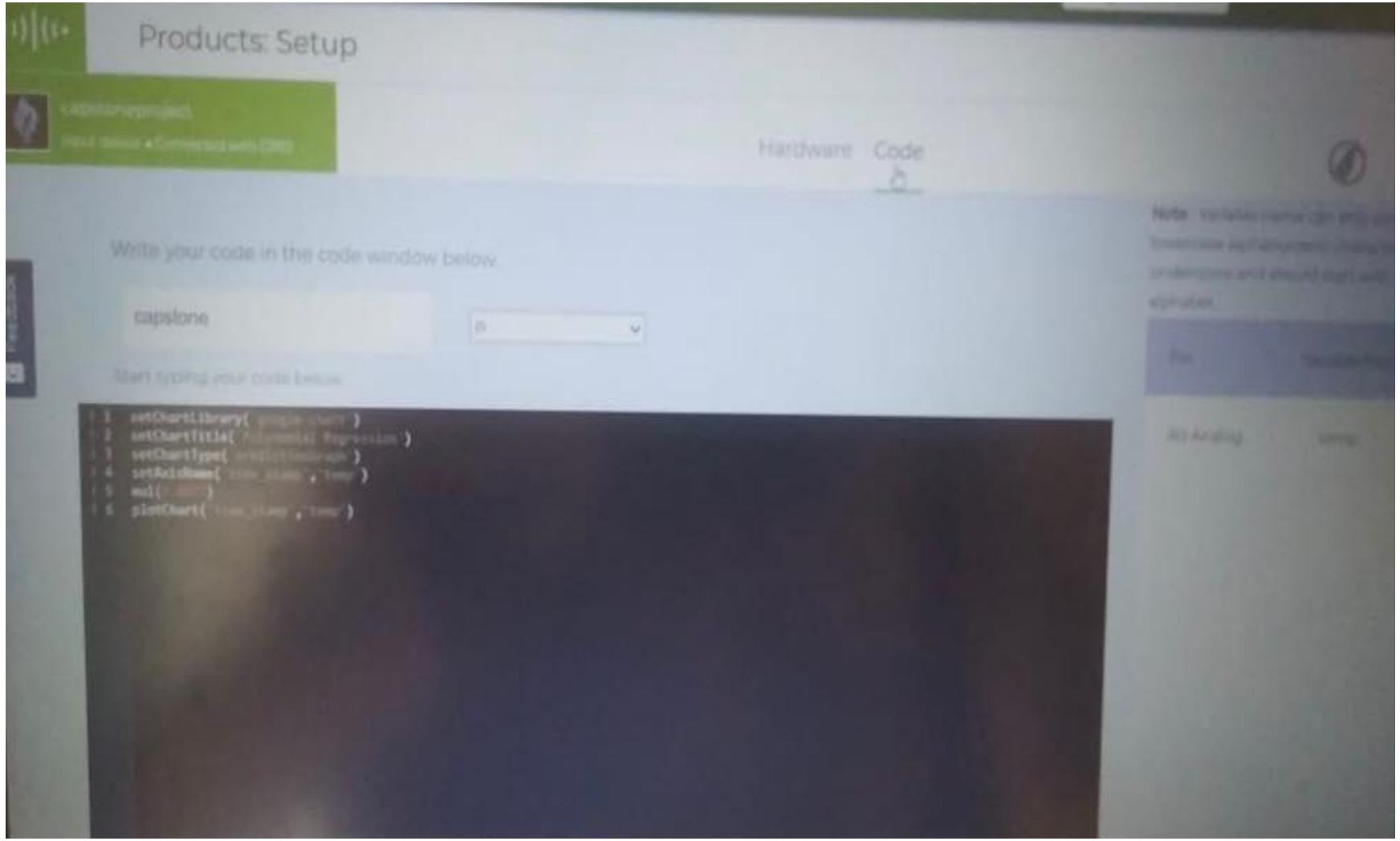
CIRCUIT CONNECTIONS

Creation of Product on BOLT Cloud

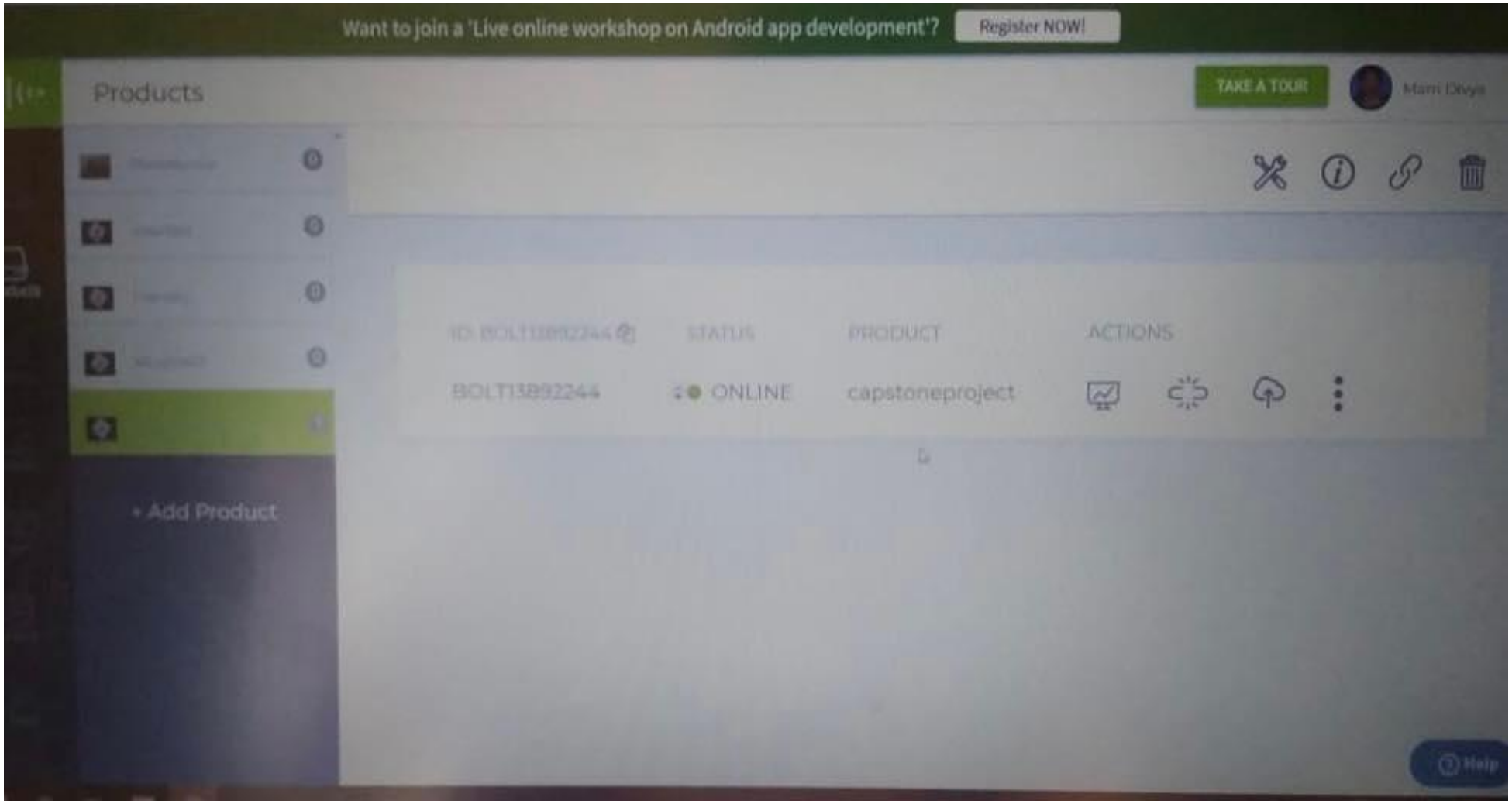



Want to join a 'Live online workshop on Android app development'?
Register NOW!
Products
TAKE A TOUR
Mani Divya
STATUS
PRODUCT
ACTIONS
BOLT13892244
ONLINE
capstoneproject
+ Add Product
Help

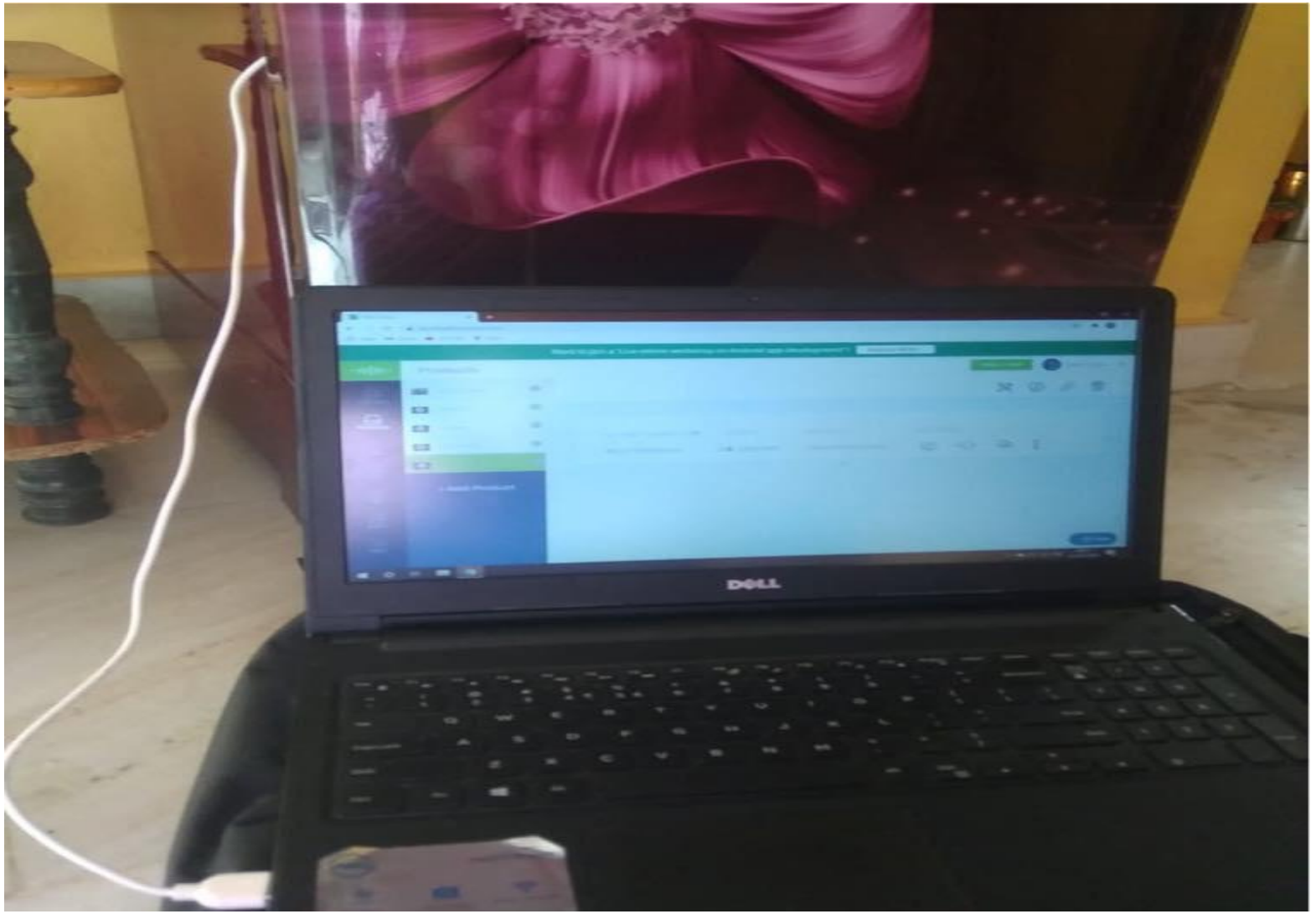
Code for Polynomial Regression



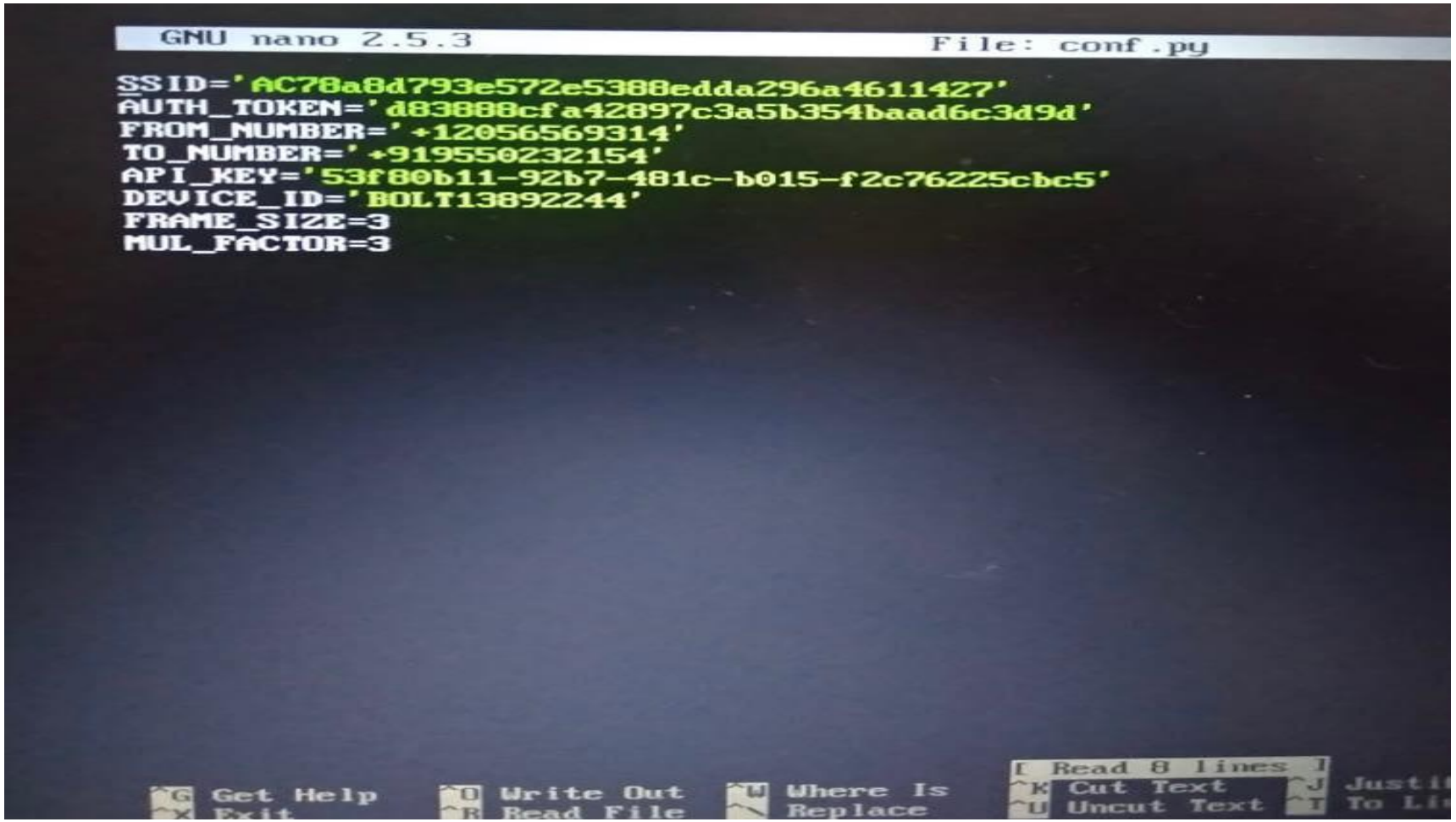
GRAPH


Want to join a 'Live online workshop on Android app development'?
Register NOW!
Products
TAKE A TOUR
STATUS
PRODUCT
ACTIONS
BOLT13892244
ONLINE
capstoneproject
+ Add Product

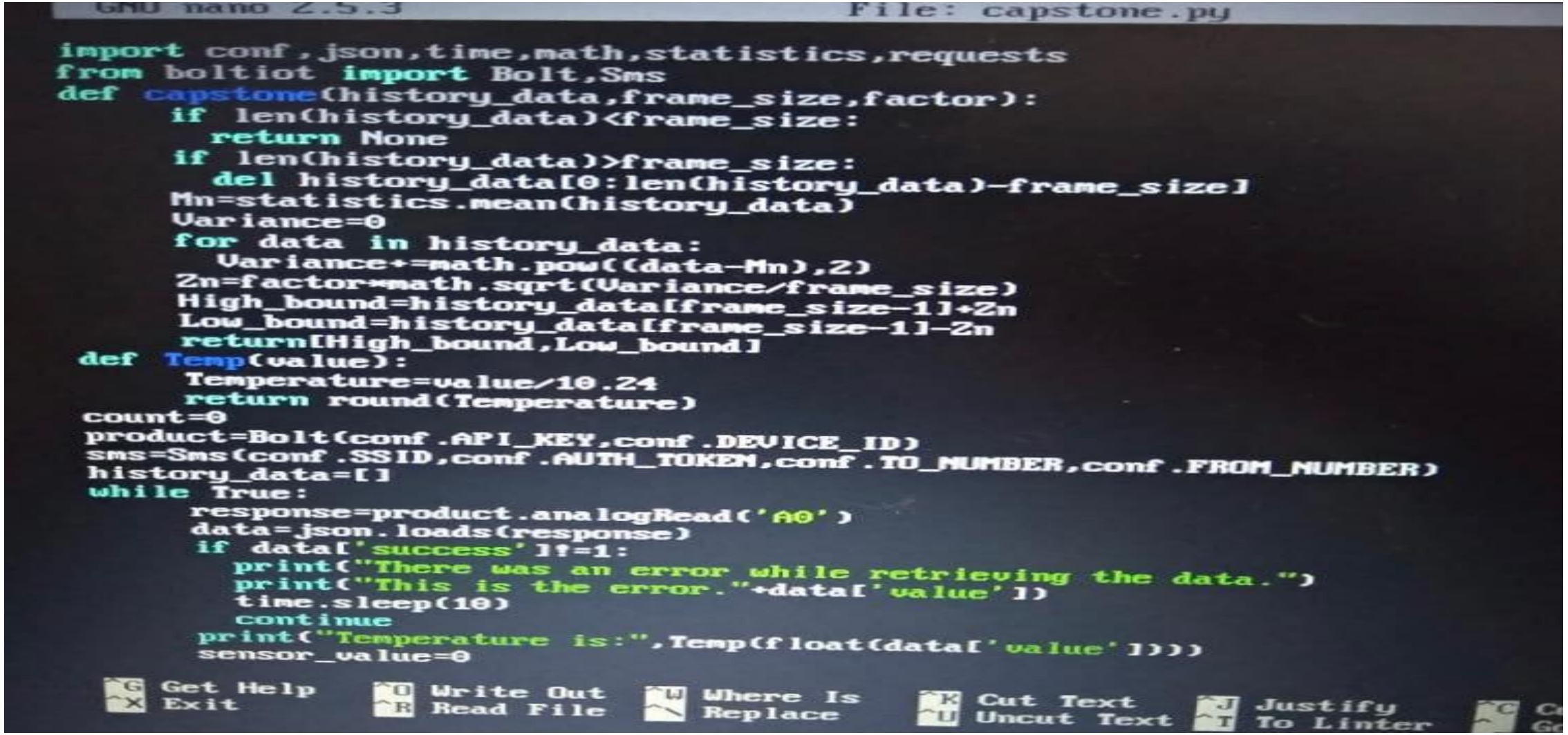
Setting up the Circuit inside the Refrigerator



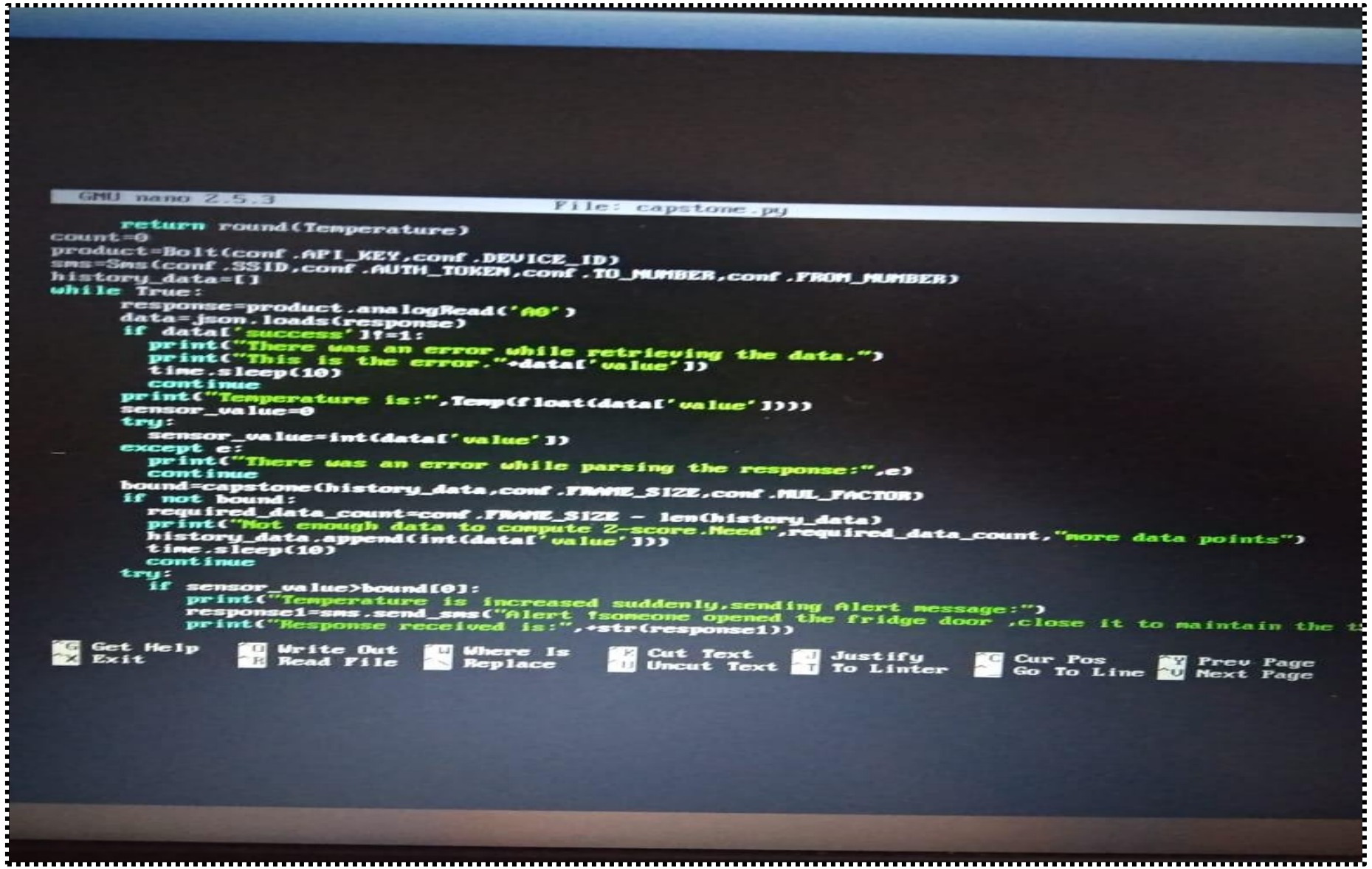
ACCESS ID’S


GNU nano 2.5.3
File: conf.py
SSID='AC78a8d793e572e5388edda296a4611427'
AUTH_TOKEN='d83888cfa42897c3a5b354baad6c3d9d'
FROM_NUMBER='+12056569314'
TO_NUMBER='+919550232154'
API_KEY='53f80b11-92b7-481c-b015-f2c76225cbc5'
DEVICE_ID='BOLT13892244'
FRAME_SIZE=3
MUL_FACTOR=3
Get Help
Write Out
Where Is
Read 8 lines
Cut Text
Uncut Text

PYTHON CODE

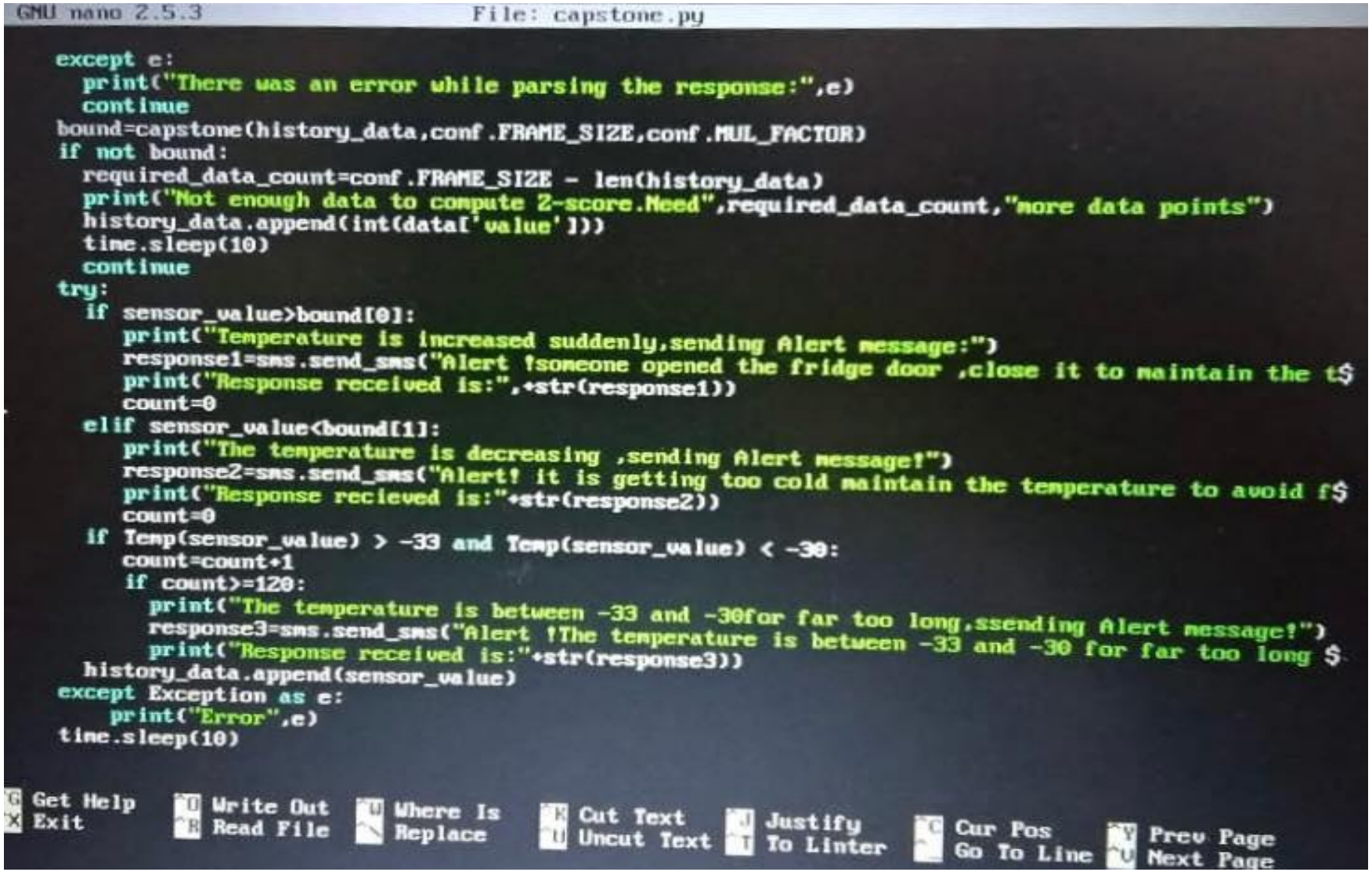

GNU nano 2.5.3 File: capstone.py
import conf,json,time,math,statistics,requests
from boltiot import Bolt,Sms
def capstone(history_data,frame_size,factor):
if len(history_data)<frame_size:
return None
if len(history_data)>frame_size:
del history_data[0:len(history_data)-frame_size]
Mn=statistics.mean(history_data)
Variance=0
for data in history_data:
Variance+=math.pow((data-Mn),2)
Zn=factor*math.sqrt(Variance/frame_size)
High_bound=history_data[frame_size-1]+Zn
Low_bound=history_data[frame_size-1]-Zn
return[High_bound,Low_bound]
def Temp(value):
Temperature=value/10.24
return round(Temperature)
count=0
product=Bolt(conf.API_KEY,conf.DEVICE_ID)
sms=Sms(conf.SSID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_NUMBER)
history_data=[]
while True:
response=product.analogRead('A0')
data=json.loads(response)
if data['success']!=1:
print("There was an error while retrieving the data.")
print("This is the error."+data['value'])
time.sleep(10)
continue
print("Temperature is:",Temp(float(data['value'])))
sensor_value=0
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter

PYTHON CODE

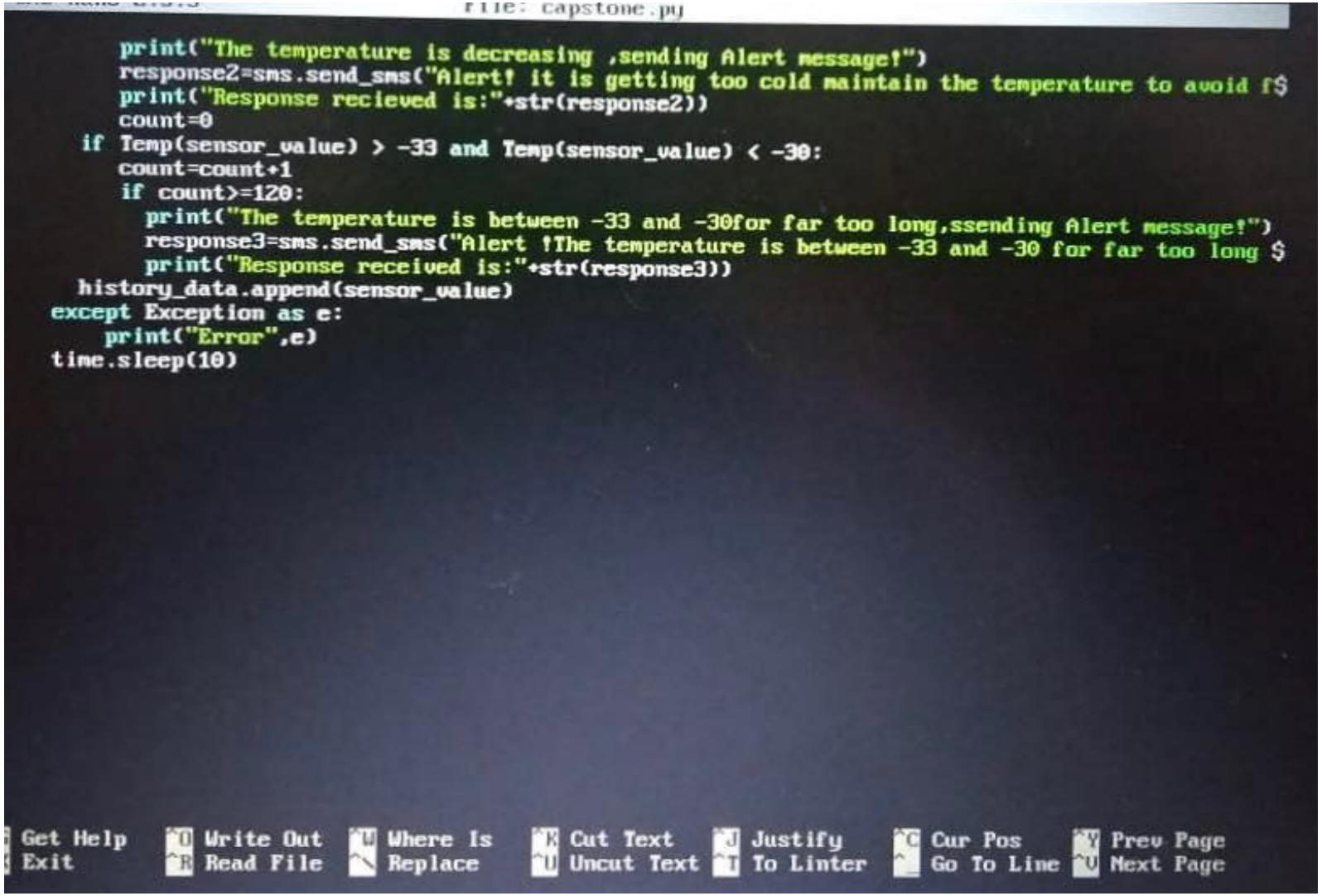


```
  GNU nano 2.5.3                    File: capstone.py

        return round(Temperature)
count=0
product=Bolt(conf.API_KEY,conf.DEVICE_ID)
sms=Sms(conf.SSID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_NUMBER)
history_data=[]
while True:
        response=product.analogRead('A0')
        data=json.loads(response)
        if data['success']!=1:
          print("There was an error while retrieving the data.")
          print("This is the error."+data['value'])
          time.sleep(10)
          continue
        print("Temperature is:",Temp(float(data['value'])))
        sensor_value=0
        try:
          sensor_value=int(data['value'])
        except e:
          print("There was an error while parsing the response:",e)
          continue
        bound=capstone(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)
        if not bound:
          required_data_count=conf.FRAME_SIZE - len(history_data)
          print("Not enough data to compute Z-score.Need",required_data_count,"more data points")
          history_data.append(int(data['value']))
          time.sleep(10)
          continue
        try:
          if sensor_value>bound[0]:
              print("Temperature is increased suddenly,sending Alert message:")
              response1=sms.send_sms("Alert !someone opened the fridge door ,close it to maintain the t
              print("Response received is:",+str(response1))

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line  ^V Next Page
```

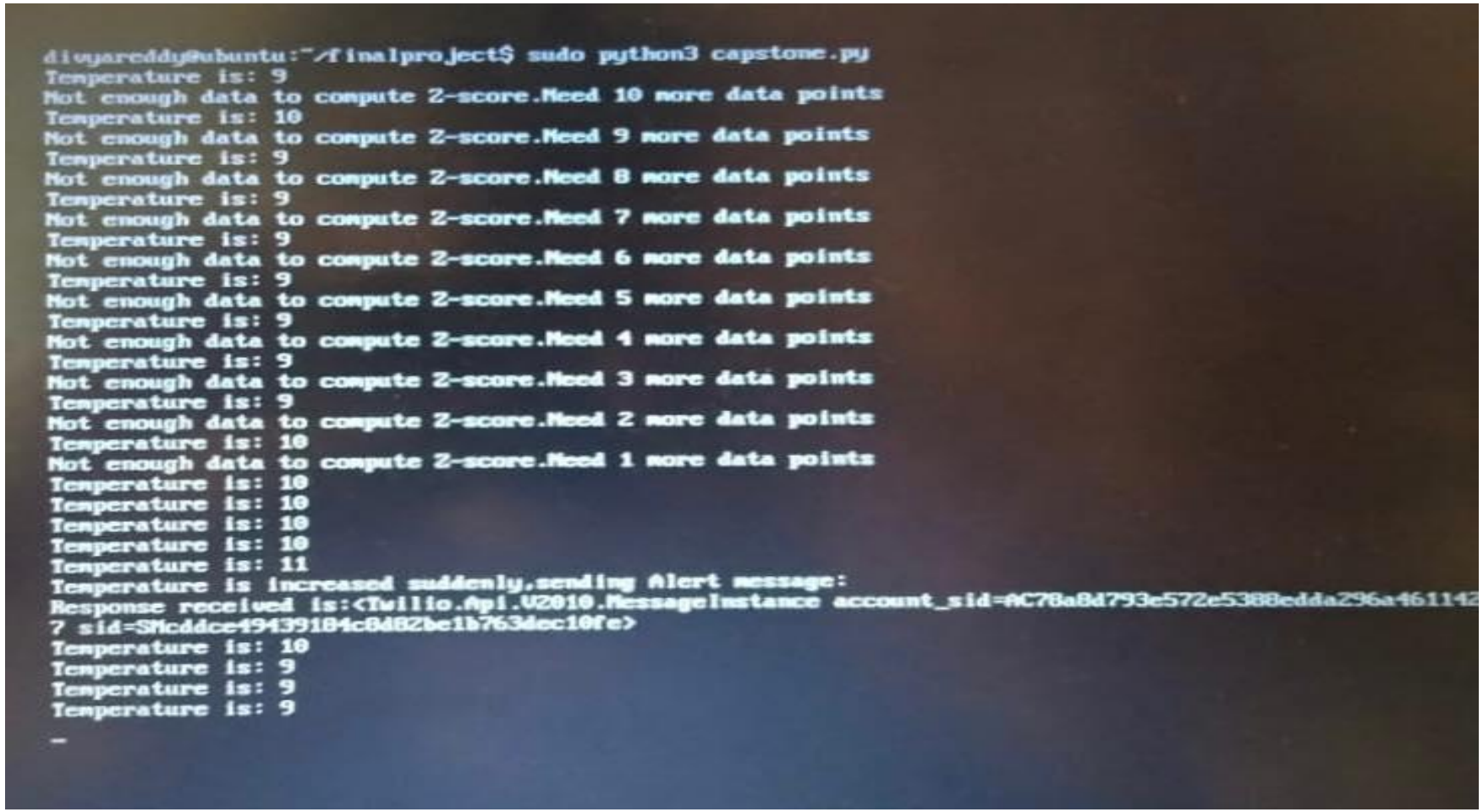
PYTHON CODE


GNU nano 2.5.3 File: capstone.py
except e:
print("There was an error while parsing the response:",e)
continue
bound=capstone(history_data,conf.FRAME_SIZE,conf.MUL_FACTOR)
if not bound:
required_data_count=conf.FRAME_SIZE - len(history_data)
print("Not enough data to compute Z-score.Need",required_data_count,"more data points")
history_data.append(int(data['value']))
time.sleep(10)
continue
try:
if sensor_value>bound[0]:
print("Temperature is increased suddenly,sending Alert message:")
response1=sms.send_sms("Alert !someone opened the fridge door ,close it to maintain the t$
print("Response received is:",+str(response1))
count=0
elif sensor_value<bound[1]:
print("The temperature is decreasing ,sending Alert message!")
response2=sms.send_sms("Alert! it is getting too cold maintain the temperature to avoid f$
print("Response recieved is:"+str(response2))
count=0
if Temp(sensor_value) > -33 and Temp(sensor_value) < -30:
count=count+1
if count>=120:
print("The temperature is between -33 and -30for far too long,ssending Alert message!")
response3=sms.send_sms("Alert !The temperature is between -33 and -30 for far too long $
print("Response received is:"+str(response3))
history_data.append(sensor_value)
except Exception as e:
print("Error",e)
time.sleep(10)
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line ^V Next Page

PYTHON CODE


File: capstone.py
print("The temperature is decreasing ,sending Alert message!")
response2=sms.send_sms("Alert! it is getting too cold maintain the temperature to avoid f$
print("Response recieved is:"+str(response2))
count=0
if Temp(sensor_value) > -33 and Temp(sensor_value) < -30:
count=count+1
if count>=120:
print("The temperature is between -33 and -30for far too long,ssending Alert message!")
response3=sms.send_sms("Alert !The temperature is between -33 and -30 for far too long $
print("Response received is:"+str(response3))
history_data.append(sensor_value)
except Exception as e:
print("Error",e)
time.sleep(10)
^G Get Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut Text
^U Uncut Text
^J Justify
^T To Linter
^C Cur Pos
^_ Go To Line
^Y Prev Page
^V Next Page

OUTPUT


divyareddy@ubuntu:~/finalproject$ sudo python3 capstone.py
Temperature is: 9
Not enough data to compute Z-score.Need 10 more data points
Temperature is: 10
Not enough data to compute Z-score.Need 9 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 8 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 7 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 6 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 5 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 4 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 3 more data points
Temperature is: 9
Not enough data to compute Z-score.Need 2 more data points
Temperature is: 10
Not enough data to compute Z-score.Need 1 more data points
Temperature is: 10
Temperature is: 10
Temperature is: 10
Temperature is: 10
Temperature is: 11
Temperature is increased suddenly,sending Alert message:
Response received is:<Twilio.Api.V2010.MessageInstance account_sid=AC78a8d793e572e5388edda296a461142
7 sid=SMcddce41439184c848Zbe1b763dec10fe>
Temperature is: 10
Temperature is: 9
Temperature is: 9
Temperature is: 9

OUTPUT ON MOBILE