SPRINT 2

|  |  |
| --- | --- |
| TEAM ID | PNT2022TMID44628 |
| PROJECT NAME | IOT BASED GADGET FOR CHILD SAFETY AND MONITORING SYSTEM |

**CONCEPT:**

Sprint 2 is about LOGIN and NOTIFIACATION of the IOT device in Parent’s Web Application for getting information about Child’s Status.

**LOGIN:**

This Coding is to build login page of parent’s application to get information about child’s condition.

**CODING:**

**Here I am using python code for get notification for parents mobile and monitoring temparature**

#!/usr/bin/python3

import RPi.GPIO as GPIO

import time

import os

import glob

from twilio.rest import Client

def main():

"""Determine the address of your temperature data on the RPi, which will always begin

with a “28.” In my case, my `device\_folder` is

‘/sys/bus/w1/devices/28-83185055b9ff/w1-slave`. That is the location on my RPi of

where the raw temperature data is being saved. The numbers following the 28 prefix

will be different for you.

"""

base\_dir = '/sys/bus/w1/devices/'

device\_folder = glob.glob(base\_dir + '28\*')[0]

device\_file = device\_folder + '/w1\_slave'

#Define your Twilio credentials

account\_sid = 'TWILIO\_ACCOUNT’

auth\_token = 'TWILIO\_TOKEN'

client = Client(account\_sid, auth\_token)

#GPIO Setup. The code needs to tell the RPi which GPIO pins to read data from.

temp\_channel = 4

temp = GPIO.setmode(GPIO.BCM)

temp = GPIO.setup(temp\_channel, GPIO.IN)

#Function to open the device file and read the raw temperature data

def read\_temp\_raw():

f = open(device\_file, 'r')

lines = f.readlines()

f.close()

return lines

#Function to extract and parse the raw temp data, and convert Celsius to Fahrenheit.

def read\_temp():

lines = read\_temp\_raw()

while lines[0].strip()[-3:] != 'YES':

time.sleep(0.2)

lines = read\_temp\_raw()

equals\_pos = lines[1].find('t=')

if equals\_pos != -1:

temp\_string = lines[1][equals\_pos+2:]

temp\_c = float(temp\_string) / 1000.0

temp\_f = temp\_c \* 9.0 / 5.0 + 32.0

temp\_f = round(temp\_f)

return temp\_f

#Function to create a text message string if the temperature is too warm.

def warm\_message():

client.messages.create(

to='ALERT\_PHONE',

from\_='TWILIO\_PHONE’,

body="It's currently " + str(read\_temp()) + " degrees in my crib, how about " \

"turning up the air conditioning or opening a window?")

#Function to create a text message string if the temperature is too cold.

def cold\_message():

client.messages.create(

to='ALERT\_PHONE',

from\_='TWILIO\_PHONE’,

body="It's currently " + str(read\_temp()) + " degrees in my crib, how about " \

"turning the heat up a little bit?")

#Run perpetually. Send the message based on the temperature.

while True:

if read\_temp() > 82:

warm\_message()

if read\_temp() < 60:

cold\_message()

time.sleep(300)

main()