ARDUINO CODE FOR ACCIDENT PREVENTION SYSTEM

#include <LiquidCrystal.h>

const int rs = 14, en = 12, d4 =11 , d5 = 10, d6 = 8,d7 = 7;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int trigPin = 2;

const int echoPin = 4;

int a=0;

const int buttonPin = 5;

const int buttonPin1 = 13;

int buttonState = 0;

int buttonState1= 0;

long duration;

int distance;

int value;

const int AOUTpin=15;//the AOUT pin of the alcohol sensor goes into analog pin A0 of the arduino

const int ledPin=12;

void setup() {

// put your setup code here, to run once:

pinMode(3,OUTPUT);

pinMode(9,OUTPUT);

pinMode(6,OUTPUT);

pinMode(13,OUTPUT);//buzzer

pinMode(2,OUTPUT); //TRIG

pinMode(4,INPUT);//ECHP

pinMode(buttonPin, INPUT);

pinMode(buttonPin1, INPUT);

pinMode(AOUTpin, INPUT\_PULLUP);

lcd.begin(16, 2);

lcd.setCursor(0,0);

lcd.print("ACC:");

lcd.setCursor(10,0);

lcd.print("Speed:");

Serial.begin(9600);

}

void loop() {

value= analogRead(AOUTpin);

Serial.println(value);//reads the analaog value from the alcohol sensor's AOUT pin

//limit= digitalRead(DOUTpin);//reads the digital value from the alcohol sensor's DOUT pin

int t=constrain(value,100,111);

t=map(t,100,111,0,100);

Serial.print("Alcohol value: ");

Serial.println(t);

//prints the alcohol value

delay(1000);

if (t<=50)

{Serial.println("DRUNK");

while(1)

{}

//normal car code

}

else{

int l=analogRead(A2);

int k=360-l;

Serial.print(k);

Serial.print('\t');

delay(500);

int m=analogRead(A3);

Serial.print(360-m);

Serial.print('\t');

delay(500);

int n=analogRead(A4);

Serial.print(360-n);

Serial.println();

if((k>2) || (k<-80))

{

lcd.setCursor(0, 1);

lcd.print("...");

for(int i=0;i<5;i++)

{

int b=digitalRead(buttonPin);

Serial.println(b);

delay(1000);

if(b==HIGH && i<5)

{

break;

}

if(b==LOW && i==4)

{

lcd.setCursor(5,0 );

lcd.print("YES");

lcd.setCursor(0,1 );

lcd.print("MSG SENT");

Serial.println("ACCIDENT");

delay(1000);

while(1)

{}

}

tone(9,356);

delay(1000);

noTone(9);

delay(1000);

}

}

else

{

Serial.println("no collision");

lcd.setCursor(0, 1);

lcd.print("NOO");

}

buttonState1= digitalRead(buttonPin1);

while(buttonState1==HIGH)

{

int l=analogRead(A2);

int k=360-l;

Serial.print(k);

Serial.print('\t');

delay(500);

int m=analogRead(A3);

Serial.print(360-m);

Serial.print('\t');

delay(500);

int n=analogRead(A4);

Serial.print(360-n);

Serial.println();

if((k>2) || (k<-80))

{

lcd.setCursor(0, 1);

lcd.print("...");

for(int i=0;i<5;i++)

{

int b=digitalRead(buttonPin);

Serial.println(b);

delay(3000);

if(b==HIGH && i<5)

{

break;

}

if(b==LOW && i==4)

{

lcd.setCursor(5,0 );

lcd.print("YES");

lcd.setCursor(0,1 );

lcd.print("MSG SENT");

Serial.println("ACC");

delay(800);

while(1)

{}

}

else

tone(9,356);

delay(1000);

noTone(9);

delay(1000);

}

tone(9,356);

delay(1000);

noTone(9);

delay(1000);

}

else

{

Serial.println("no collision");

lcd.setCursor(0, 1);

lcd.print("NOO");

}

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance= duration\*0.034/2;

//Serial.print("Distance: ");

Serial.println(distance);

int s = map(a,0,255,0,100);

//Serial.println(s);

//lcd.setCursor(0, 1);

//lcd.print(distance);

//lcd.print("%");

delay(75);

lcd.print(" ");

if (distance>=20 && distance<35)

{

a=154;

analogWrite(3,0);

analogWrite(6,a);

Serial.println("hi");

lcd.setCursor(13, 1);

lcd.print("%");

lcd.setCursor(10, 1);

lcd.print(s);

}

if (distance>35)

{

a=255;

analogWrite(3,0);

analogWrite(6,a);

lcd.setCursor(10, 1);

lcd.print(s);

}

if (distance<20)

{

a=0;

analogWrite(3,a);

analogWrite(6,a);

lcd.setCursor(10, 1);

lcd.print(s);A

}

}}

}

#PYTHON CODE FOR ALERTING SYSTEM(MESSAGING,CALCULATING NEAREST HOSPITAL,DETECTING CURRENT LOCATION

# LIBRARIES

# for creating pop boxes

from tkinter import \*

from tkinter import messagebox

# for system functions

import sys

# for keyboard automatiom

import pyautogui as py

# for web automation

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

from selenium.webdriver.support.ui import WebDriverWait

# for delays,date,&others

import time

#for arduino-pi(computer) serial communication

import serial

# IMAGE CAPTURING

import cv2

# for creating dataframes from .csv file

import pandas as pd

#for calculating distance b/w locations

from geopy.distance import geodesic as geo

#for email sytem

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from email.mime.base import MIMEBase

from email import encoders

from email.mime.image import MIMEImage

# messaging API FAST2SMS

import requests

#global variables

latitude=0

longitude=0

s=serial.Serial('/dev/cu.usbmodem1431',9600) # serial object

email\_user = 'yyyy@gmail.com' # sender email id

email\_send= 'xxxx@gmail.com' # reciever email id

email\_password = 'aaxxccsd' #sender id password

def currentloc(): #for calculating current location of accident

global latitude,longitude

chrome\_options = Options()

chrome\_options.add\_argument("--use-fake-ui-for-media-stream")

timeout = 30

print('\007')

driver = webdriver.Chrome("/Users/harsimranwadali/Documents/chromedriver")

driver.get("https://mycurrentlocation.net/")

wait = WebDriverWait(driver, timeout)

time.sleep(2)

longitude = driver.find\_elements\_by\_xpath('//\*[@id="longitude"]')

longitude = [x.text for x in longitude]

time.sleep(2)

longitude = str(longitude[0])

latitude = driver.find\_elements\_by\_xpath('//\*[@id="latitude"]')

latitude = [x.text for x in latitude]

latitude = str(latitude[0])

time.sleep(2)

driver.quit()

def nearestdist(origin): #for finding nearest hospital from current location

df1=pd.read\_csv("database2.csv")

print(df1)

lats=list(df1['LATITUDE'])

longs=list(df1['LONGITUDE'])

names=list(df1['NAME'])

phonenos=list(df1['PHONENOS'])

l=len(lats)

lis=list(zip(lats,longs))

d=[]

print('\007')

messagebox.showinfo("DISTANCE CALCULATION","CALCULATING NEAREST HOSPITAL LOCATION FROM HOSPITALS DATABASE!!....")

a=lambda x:(geo(origin,x).miles)

d=list(map(a,lis))

d = [ round(elem, 2) for elem in d ]

print(d)

s=min(d)

i=0

while i<l:

if s==d[i]:

print("THE NEAREST HOSPITAL IS ",names[i])

print('\007')

messagebox.showinfo(" NEAREST HOSPITAL FOUND ",'THE NEAREST HOSPITAL IS '+names[i])

time.sleep(2)

print("ITS DISTANCE FROM CURRENT LOCATION IS "+str(s)+" miles")

print('\007')

messagebox.showinfo("NEAREST DISTANCE","ITS DISTANCE FROM CURRENT LOCATION IS "+str(s)+" miles")

time.sleep(2)

sms(names[i],origin,phonenos[i])

time.sleep(3)

print('\007')

showloc(origin,(lats[i],longs[i]))

break

i=i+1

sys.exit()

def sms(name,origin,phone): # For sending the SMS notifications along with location link to the nearest Hospital.

apikey='KB7uowO9J1MlnicE3hZTFaSr40geIvUy6QmHt2dfqNsGRp8VzCACxgOJ2GQksKfiwb5S1u9nXMF6LzDI'

url = "https://www.fast2sms.com/dev/bulk"

message="CALL TO "+name+" FOR HELP \n reach asap"

querystring = {"authorization":apikey,"sender\_id":"FSTSMS","message":message,"language":"english","route":"p","numbers":str(phone)}

headers = {

'cache-control': "no-cache"

}

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

print(response.text)

message1="ACCIDENT OCCURS AT THIS LOCATION--\n https://www.google.com/maps?q="+str(origin[0])+','+str(origin[1]) #msg1 to be sent

querystring = {"authorization":apikey,"sender\_id":"FSTSMS","message":message1,"language":"english","route":"p","numbers":str(phone)}

headers = {

'cache-control': "no-cache"

}

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

print(response.text)

messagebox.showinfo('SMS STATUS','SENDING SMS TO THE NEAREST HOSPITAL('+name+')') # msg2 to be sent

time.sleep(2)

print("SMS SENT")

time.sleep(1)

print('\007')

messagebox.showinfo('SMS STATUS','SMS SENT SUCCESSFULLY TO THE NEAREST HOSPITAL')

time.sleep(2)

def showloc(origin,destination):# for opening googlemap link which showing the path b/w nearest hospital and the location where accident occurs

print(origin)

print(destination)

driver = webdriver.Chrome("/Users/harsimranwadali/Documents/chromedriver")

url='https://www.google.com/maps/dir/?api=1&origin='+str(origin[0])+','+str(origin[1])+"&destination="+str(destination[0])+','+str(destination[1])

driver.get(url)

time.sleep(2)

py.typewrite(['enter'])

time.sleep(10)

driver.quit()

def imagecapture(): # for capturing image of a Driver when alcohol content greater than legal limit is found before the car's ignition

data= time.strftime("%d\_%b\_%Y|%H:%M:%S")

print(data)

camera = cv2.VideoCapture(0)

return\_value, image = camera.read()

cv2.imwrite(str(data)+'.jpg', image)

del(camera)

messagebox.showinfo("DRUNK AND DRIVE CASE",'IMAGE OF DRUNKEN DRIVER CAPTURED')

print('\007')

gmail(data)

def gmail(data): #for sending email through gmail along with captured photo of driver

mail = MIMEMultipart()

mail['From'] =email\_user

mail['To'] = email\_send

mail['Subject'] = "ALERT ALERT!!"

body = "THE DRIVER DRIVING VEHICLE NO-PB02XXXX IS TOTALLY DRUNKEN .ALSO, PHOTO OF THE DRIVER IS ATTACHED"

mail.attach(MIMEText(body, 'plain'))

print (data)

dat='%s.jpg'%data

print (dat)

attachment = open(dat, 'rb')

image=MIMEImage(attachment.read())

attachment.close()

mail.attach(image)

server = smtplib.SMTP('smtp.gmail.com', 587)

server.starttls()

server.login(email\_user,email\_password )

text = mail.as\_string()

server.sendmail(email\_user,email\_send, text)

messagebox.showinfo('DRUNK AND DRIVE CASE','EMAIL HAS BEEN SUCCESFULLY SENT ALONG WITH IMAGE')

print('\007')

time.sleep(2)

server.quit()

#THE CODE EXECUTION STARTS FROM HERE

while 1:

if x=='DRUNK: #signal received serially from Arduino when Alcohol consumption is sensed by Alcohol sensor

print('\007')

root=Tk()

root.title('ALERTING SYSTEM')

root.geometry('500x300+200+300')

label=Label(root,text="IMAGE CAPTURING \n\n SENDING EMAIL THROUGH GMAIL ",bg='red')

label.pack(fill=X)

messagebox.showinfo('ALCOHOL SENSOR','SIGNAL DETECTED FROM SENSOR')

time.sleep(2)

imagecapture()

root.destroy

sys.exit()

else:

a=s.readline().decode() #Read the data send by arduino serially and decode it as string value

print(a)

if 'ACCIDENT' in a: #signal received from Arduino when Accident occurrence has been sensed by different sensors on the vehicle.

print('\007')

root=Tk()

root.title('ALERTING SYSTEM')

root.geometry('500x300+200+300')

label=Label(root,text="FINDING CURRENT LOCATION USING WEB AUTOMATION\n CALCULATING NEAREST HOSPITAL FROM DATABASE\nSENDING SMS TO THE NEAREST HOSPITAL\nSHOW THE PATH B/W NEAREST HOSPITAL AND ACCIDENT LOCATION",bg='red')

label.pack(fill=X)

messagebox.showinfo('ACCIDENT HAPPENS','COLLISION SIGNAL DETECTED')

time.sleep(2)

currentloc()

origin=(latitude,longitude)

messagebox.showinfo("CURRENT LOCATION","LATITUDE--"+str(latitude)+"\nLONGITUDE---"+str(longitude))

origin1=(float(origin[0]),float(origin[1]))

nearestdist(origin1)

root.destroy

sys.exit()