**SCHOOL OF COMPUTING (SOC)**

|  |  |
| --- | --- |
| **Date of Submission:** | 21/2/2021 |

|  |  |
| --- | --- |
| **Prepared for:** | Ms Dora Chua |

|  |  |
| --- | --- |
| **Class:** | DISM/FT/3A/41 |

|  |  |
| --- | --- |
| **Submitted by:** |  |

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 1828261 | Chee Poh Chuan |
| 1828089 | Vincent Sow Zhao Bin |
|  |  |

**IOT CA2**

**Step-by-step Tutorial**

**DIPLOMA IN BUSINESS INFORMATION TECHNOLOGY**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**DIPLOMA IN INFOCOMM SECURITY MANAGEMENT**

**ST0324 Internet of Things (IOT)**

**2017/2018 Semester 1**

**Table of Contents**

[Section 1 Overview of project 3](#_Toc64825153)

[A. Where we have uploaded our tutorial 3](#_Toc64825154)

[B. What is the application about? 3](#_Toc64825155)

[C. How does the final RPI set-up looks like? 4](#_Toc64825156)

[D. How does the web or mobile application look like? 5](#_Toc64825157)

[E. System architecture of our system 6](#_Toc64825158)

[F. Evidence that we have met basic requirements 6](#_Toc64825159)

[G. Bonus features on top of basic requirements 7](#_Toc64825160)

[A. Quick-start guide (Readme first) 7](#_Toc64825161)

[Section 2 Hardware requirements 8](#_Toc64825162)

[Hardware checklist 8](#_Toc64825163)

[Fritzing Diagram 8](#_Toc64825164)

[Section 3 Software Requirements 9](#_Toc64825165)

[Software checklist 9](#_Toc64825166)

[Software setup instructions 9](#_Toc64825167)

[AWS Services set up 10](#_Toc64825168)

[Registering the raspberry pi as a thing. 10](#_Toc64825169)

[Create Security Policy for the RPi 11](#_Toc64825170)

[Attaching Security policy and thing to your Cert 11](#_Toc64825171)

[REST API Endpoint for your thing 11](#_Toc64825172)

[Create Role 12](#_Toc64825173)

[Create DynamoDB table 12](#_Toc64825174)

[Create DynamoDB table rule 12](#_Toc64825175)

[Create AWS credentials file 13](#_Toc64825176)

[Create and confifure a bucket on Amazon S3 14](#_Toc64825177)

[Section 4 Source codes 15](#_Toc64825178)

[server.py 15](#_Toc64825179)

[ldr-mqtt.py 18](#_Toc64825180)

[jsonconverter.py 19](#_Toc64825181)

[dynamodb.py 20](#_Toc64825182)

[detect\_motion.py 21](#_Toc64825183)

[index.html 24](#_Toc64825184)

[Section 5 Task List 38](#_Toc64825185)

[Section 6 References 38](#_Toc64825186)

# Section 1 Overview of project

* 1. Where we have uploaded our tutorial

Fill up the Google form here to submit your links and then paste the links here of your Youtube and tutorial document here as well.

<http://bit.ly/1910s2iotca2>

|  |  |
| --- | --- |
| **Youtube** | https://youtu.be/0ZbBpmvjZ-w |
| **Public tutorial link** | https://github.com/cheepohchuan/IOT-CA2 |

* 1. What is the application about?

The Smart Room Security System is an application that utilizes multiple sensors to allow homeowners to view and manage their room enviroment, as well as home security. Allowing the homeowners to view the current temperature and humidity in their room, as well as view the light levels in their room. Secondly, it has a smart security alert system that uses a motion sensor and a camera to identify unwanted presence in the room using image recognition.

The target audience for this application is any homeowners that want to view the room enviromental status as well as a smart alert system to notify them of possible intrusions.

The application uses sensors to monitor the enviromental conditions in the room to allow the homeowners to know if the room is getting too hot or too cold, as well as the light levels to see if a room is too dark. Also, the smart security alert system can help homeowners who are worried about possible intrusions.

* 1. How does the final RPI set-up looks like?

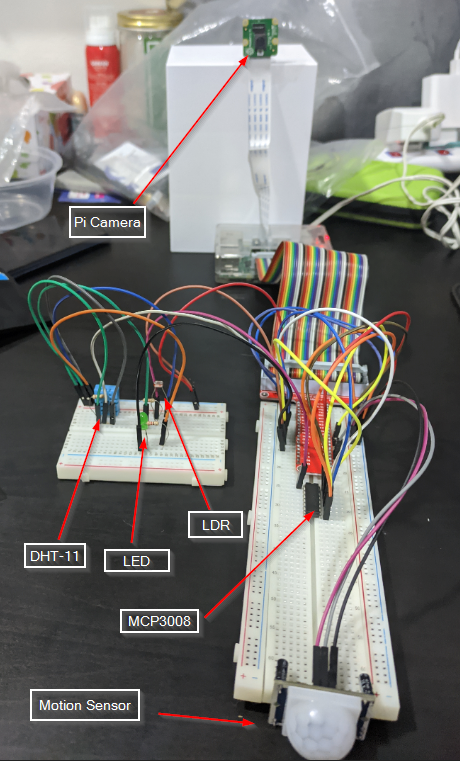


Figure 1.1 Final Hardware setup

* 1. How does the web or mobile application look like?

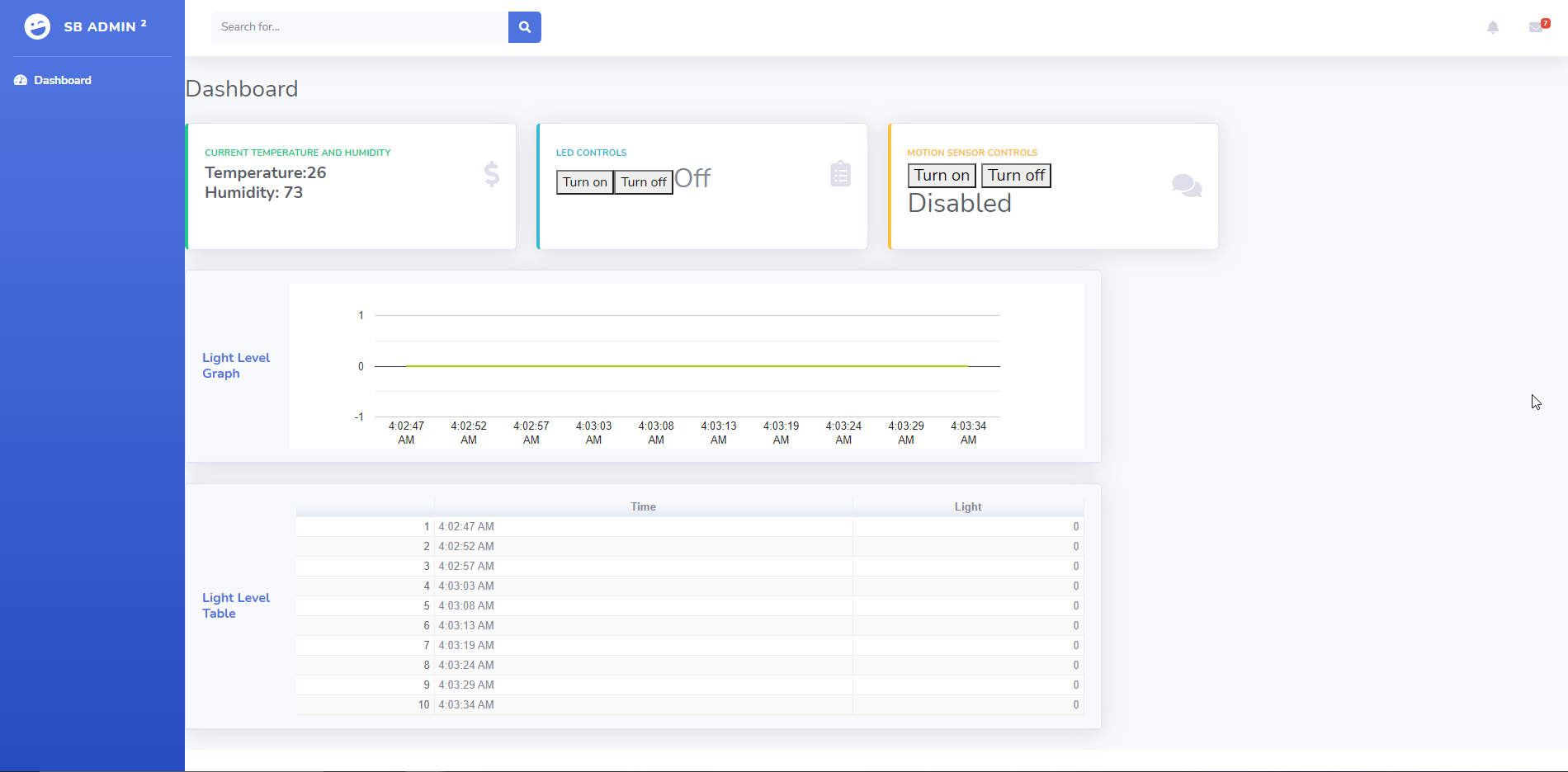
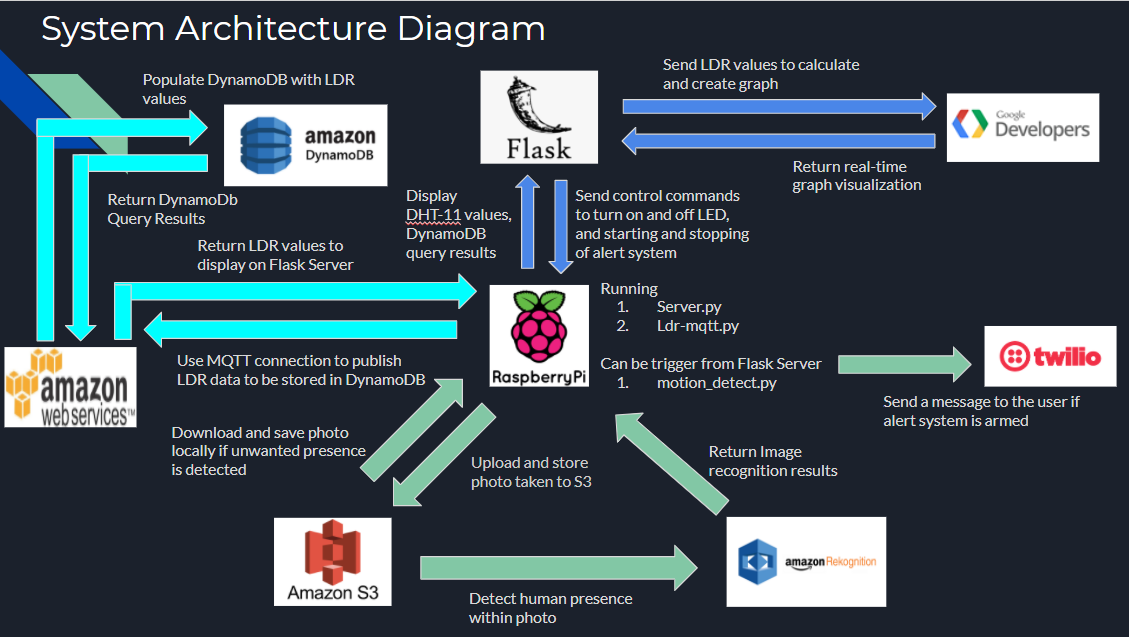


Figure 1.2 Flask Server index.html

* 1. System architecture of our system



* 1. Evidence that we have met basic requirements

|  |  |
| --- | --- |
| Requirement | Evidence |
| Used three sensors | Used:   * + 1. LED     2. LDR     3. DHT-11     4. Pir Motion Sensor |
| Used MQTT | Our MQTT endpoint -->  Example of data sent through MQTT : ??? |
| Stored data in cloud | Stored light data in AWS DynamoDB |
| Used cloud service | Used AWS Rekognition and AWS S3 |
| Provide real-time sensor value / status | Show the real-time value of DHT-11 sensor |
| Provide historical sensor value/ status | Show historical vlaue of LDR sensor |
| Control actuator | Placed button on webpage to control LED |

* 1. Bonus features on top of basic requirements

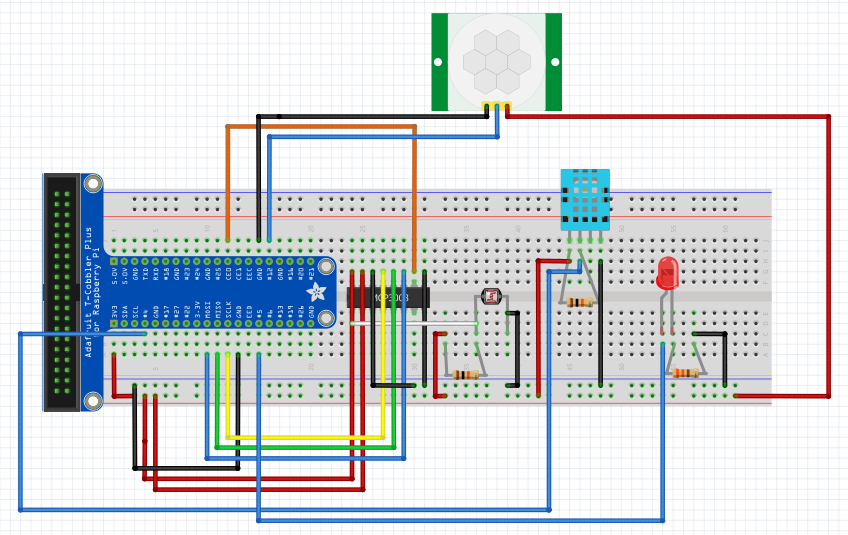
1. SMS to alert user of unwanted presence in the room
   1. Quick-start guide (Readme first)
2. First connect hardware as in Section 2
3. Then run the server.py file for web server
4. Run the ldr-mqtt.py to publish light values to AWS using MQTT

# Section 2 Hardware requirements

Hardware checklist

* + 1. Raspberry Pi
    2. MCP3008
    3. Pir Motion Sensor
    4. LED
    5. LDR
    6. DHT-11
    7. Resisters
    8. Pi Camera

Fritzing Diagram

****

# Section 3 Software Requirements

Software checklist

1. Flask
2. Gevent
3. Boto3
4. Numpy
5. Botocore

Software setup instructions

AWS Python Library. The AWS Python Libraries are requires before the program can interact with AWS.

sudo pip install --upgrade --force-reinstall pip==9.0.3

sudo pip install AWSIoTPythonSDK --upgrade --disable-pip-version-check

sudo pip install --upgrade pip

awscli.

Sudo pip install awscli

If awscli is already installed and you want to upgrade it.

Sudo pip install awscli –upgrade

Botocore.

Sudo pip install botocore

If botocore is already installed, but you want to upgrade it.

Sudo pip install botocore –upgrade

Sudo pip install boto3 –upgrade

For the other packages if its not install just run.

Sudo pip install flask gevent numpy

AWS Services set up

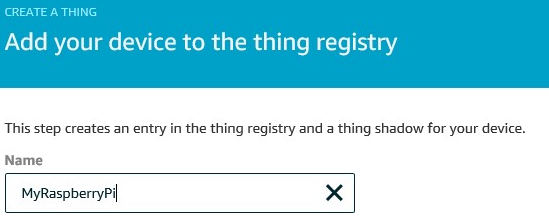
## Registering the raspberry pi as a thing.

1. Search for "IOT Core" and click it to access the IOT Core dashboard.

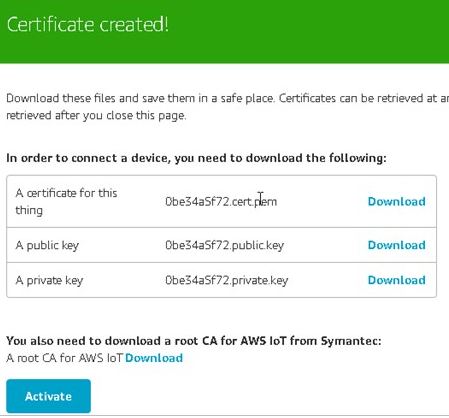
2. In the left navigation pane, click “Manage” to expand it, then choose “Things”.

3. Choose "Register a thing" or "Create" to create a new Thing.

4. Name your Thing and click "Create a single thing".



5. Choose “One-click certificate creation” to generate acertificate and key pair. Download the certificate, public key and private key. For the root CA, you can download at this link: <https://docs.aws.amazon.com/iot/latest/developerguide/server-authentication.html#server-authentication-certs>. Right click on Amazon Root CA 1 and click "Save As".



6. Store the certificates in the same folder as server.py and database.py. Rename the certificates to friendlier names if required. Next, click the “Activate” button. You should see “Successfully activated certificate”.

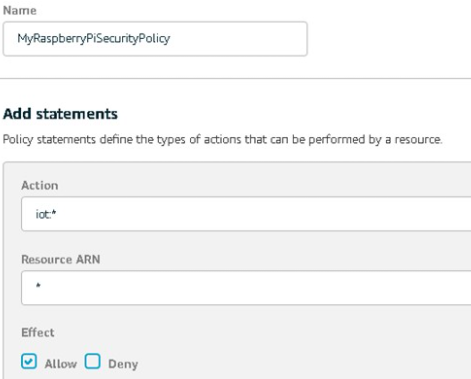
7. Click to the next page, and click “Register Thing”.

## Create Security Policy for the RPi

1. On the left IOT Core dashboard, select Policies under the Secure menu.

2. On the next page, choose “Create new policy”.

3. Key in the following configuration with a policy name of your choice:



4. Press “Create”

## Attaching Security policy and thing to your Cert

1. On the left IOT Core dashboard, select Certificates under the Secure sub-menu.

2. Click on the certificate. Under Actions, click "Attach Policy" and choose the policy you created earlier. Click the "Attach" button.

3. Under Actions, click "Attach Thing" and select the checkbox next to the Thing you created. Click "Attach".

## REST API Endpoint for your thing

Click “Manage->Things” and choose your Thing. On the next screen, choose “Interact”. Copy and note down the REST API endpoint, as you will need it to replace the endpoints in the python code.

## Create Role

1. Search for the IAM service on AWS Console and choose “Roles”.

2. Click “Create Role” and on the next page, choose “AWS service”, then “IOT”.

3. Under “Select your use case”, select IoT.

4. Click “Next->Permissions”, click “Next->Tags”, click “Next->Review”.

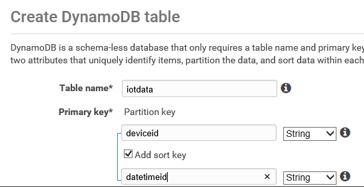
5. You will see a page that requires you to input a name for your Role. Key in a role name of your choosing.

## Create DynamoDB table

1. Create a DynamoDB table for the LDR Values

a) Open the Amazon DynamoDB Console and click on “Create Table”

b) follow the image below:



## Create DynamoDB table rule

1. In the AWS IoT console, in the left navigation pane, choose “Act”, then “Create a rule”.

2. On the Create a rule page, in the Name field, type a name for your rule e.g **MyDynamoDBRule**. In the Description field, type a description for the rule.

3. Scroll down to Rule Query statement. Type SELECT \* FROM 'sensors/light'.

4. In Set one or more actions, choose Add action.

5. On the Select an action page, select the action below. Next, choose Configure action.



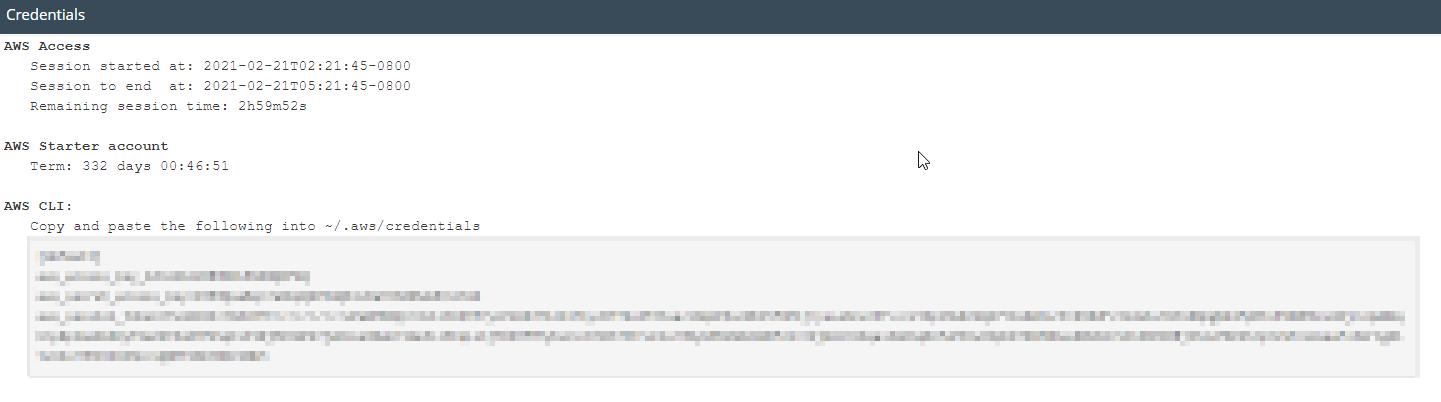
6. On the Configure action page, from the SNS target drop-down list, choose the DHT table you created earlier. g) Under **IAM role name**, choose the one you created earlier from the drop-down list and click “Update Role”.

7. Click “Create” then "Create Rule".

## Create AWS credentials file

The application uses Amazon S3 bucket as well as AWS Rekognition, and for these to work, we need to set up the credentails file in the RPi.

1. On your AWS Account Status, Click “Account Details” button. You will be shown a screen similar to this. Copy the AWS CLI code.



2. In the RPi, open a terminal windows and type:

1. sudo rm ~/.aws/credentials

2. sudo nano ~/.aws/credentials

3. Paste the code into the editor and press Ctrl-O, Ctrl-X to save.

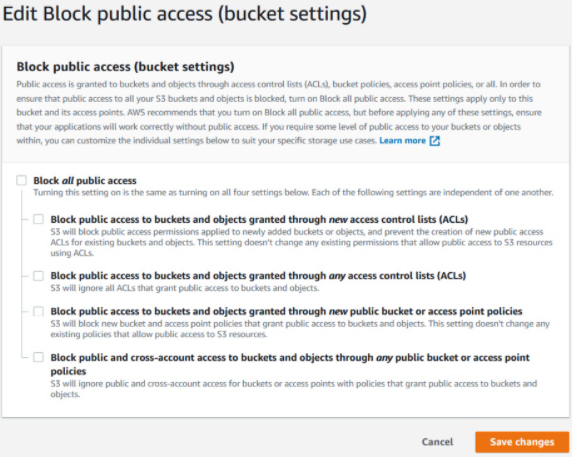
## Create and confifure a bucket on Amazon S3

To store images from the RPi, the Amazon S3 Bucket permissions and policy must be configured.

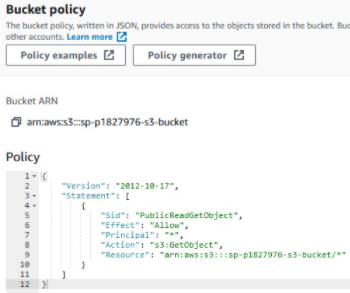
1. In the AWS console, search for S3. Click "Create bucket".

2. Type in a unique name for your bucket and choose a suitable region.

3. Click the "Permissions" tab of your newly created bucket. Click the "Edit" button under **Block public access (bucket settings)**. Uncheck everything like shown, and click "Save changes".



4. Click the "Edit" button under **Bucket policy**, and copy this code into **Policy**, with the resource name changed to your bucket. Click "Save changes".



# Section 4 Source codes

All source codes, including Python, HTML files etc

### server.py

from flask import Flask, render\_template, jsonify, request,Response

import mysql.connector

import sys

import json

import numpy

import datetime

import decimal

import gevent

import gevent.monkey

from gevent.pywsgi import WSGIServer

#from picamera.array import PiRGBArray

#from picamera import PiCamera

import cv2

import socket

import io

#camera = PiCamera()

#rawCapture = PiRGBArray(camera)

host = "a159h7ypk3cmca-ats.iot.us-east-1.amazonaws.com"

rootCAPath = "AmazonRootCA1.pem"

certificatePath = "cert.crt"

privateKeyPath = "4a2af4428b-private.pem.key"

gevent.monkey.patch\_all()

class GenericEncoder(json.JSONEncoder):

    def default(self, obj):

        if isinstance(obj, numpy.generic):

            return numpy.asscalar(obj)

        elif isinstance(obj, datetime.datetime):

            return obj.strftime('%Y-%m-%d %H:%M:%S')

        elif isinstance(obj, decimal.Decimal):

            return float(obj)

        else:

            return json.JSONEncoder.default(self, obj)

app = Flask(\_\_name\_\_)

import dynamodb

import jsonconverter as jsonc

@app.route("/api/getdata",methods=['POST','GET'])

def apidata\_getdata():

    if request.method == 'POST' or request.method == 'GET':

        try:

            data = {'chart\_data': jsonc.data\_to\_json(dynamodb.get\_data\_from\_dynamodb()),

             'title': "IOT Data"}

            return jsonify(data)

        except:

            import sys

            print(sys.exc\_info()[0])

            print(sys.exc\_info()[1])

@app.route("/")

def chartsimple():

    return render\_template('index.html')

from gpiozero import LED

led = LED(5)

def ledOn():

  led.on()

  return "On"

def ledOff():

  led.off()

  return "Off"

import os

#change this to use aws rekognition

def motion\_start():

    os.system('python detect\_motion.py &')

    return "Enabled"

def motion\_stop():

    try:

        os.system("pkill -9 -f detect\_motion.py")

        return "Disabled"

    except:

        return "Error"

@app.route("/motion/<status>")

def motion(status):

    if status == "Start":

        response = motion\_start()

    else:

        response = motion\_stop()

    return response

@app.route("/writeLED/<status>")

def writePin(status):

   if status == 'On':

     response = ledOn()

   else:

     response = ledOff()

   return response

import Adafruit\_DHT

pin = 4

@app.route("/getDHT",methods = ['POST', 'GET'])

def getDHT():

    humidity, temperature = Adafruit\_DHT.read\_retry(11, pin)

    #import random

    #humidity, temperature = random.randint(1,100), random.randint(20,30)

    data = {'temperature': temperature, 'humidity': humidity}

    print(data)

    return jsonify(data)

"""

@app.route('/live')

def index():

    return render\_template('live.html')

def gen():

    while True:

        camera.capture(rawCapture, format="bgr")

        image = rawCapture.array

        cv2.imwrite("t.jpg", image)

        rawCapture.truncate(0)

        yield (b'--frame\r\n'

               b'Content-Type: image/jpeg\r\n\r\n' + open('t.jpg', 'rb').read() + b'\r\n')

@app.route('/video')

def video():

    return Response(gen(),

                    mimetype='multipart/x-mixed-replace; boundary=frame')

"""

if \_\_name\_\_ == '\_\_main\_\_':

   try:

        print('Server waiting for requests')

        http\_server = WSGIServer(('0.0.0.0', 8001), app)

        app.debug = True

        http\_server.serve\_forever()

   except:

        print("Exception")

        import sys

        print(sys.exc\_info()[0])

        print(sys.exc\_info()[1])

### ldr-mqtt.py

# Import SDK packages

from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

from time import sleep

from gpiozero import MCP3008

adc = MCP3008(channel=0)

# Custom MQTT message callback

def customCallback(client, userdata, message):

    print("Received a new message: ")

    print(message.payload)

    print("from topic: ")

    print(message.topic)

    print("--------------\n\n")

host = "a159h7ypk3cmca-ats.iot.us-east-1.amazonaws.com"

rootCAPath = "AmazonRootCA1.pem"

certificatePath = "cert.crt"

privateKeyPath = "4a2af4428b-private.pem.key"

my\_rpi = AWSIoTMQTTClient("PubSub-1828261")

my\_rpi.configureEndpoint(host, 8883)

my\_rpi.configureCredentials(rootCAPath, privateKeyPath, certificatePath)

my\_rpi.configureOfflinePublishQueueing(-1)  # Infinite offline Publish queueing

my\_rpi.configureDrainingFrequency(2)  # Draining: 2 Hz

my\_rpi.configureConnectDisconnectTimeout(10)  # 10 sec

my\_rpi.configureMQTTOperationTimeout(5)  # 5 sec

# Connect and subscribe to AWS IoT

my\_rpi.connect()

my\_rpi.subscribe("sensors/light", 1, customCallback)

sleep(2)

# Publish to the same topic in a loop forever

loopCount = 0

while True:

    light = round(1024-(adc.value\*1024))

    loopCount = loopCount+1

    message = {}

    message["deviceid"] = "deviceid\_dorachua"

    import datetime as datetime

    now = datetime.datetime.now()

    message["datetimeid"] = now.isoformat()

    message["value"] = light

    import json

    my\_rpi.publish("sensors/light", json.dumps(message), 1)

    sleep(5)

### jsonconverter.py

from decimal import Decimal

import json

import datetime

import numpy

class GenericEncoder(json.JSONEncoder):

    def default(self, obj):

        if isinstance(obj, numpy.generic):

            return numpy.asscalar(obj)

        elif isinstance(obj, Decimal):

            return str(obj)

        elif isinstance(obj, datetime.datetime):

            return obj.strftime('%Y-%m-%d %H:%M:%S')

        elif isinstance(obj, Decimal):

            return float(obj)

        else:

            return json.JSONEncoder.default(self, obj)

def data\_to\_json(data):

    json\_data = json.dumps(data,cls=GenericEncoder)

    print(json\_data)

    return json\_data

### dynamodb.py

def get\_data\_from\_dynamodb():

    try:

            import boto3

            import datetime

            from boto3.dynamodb.conditions import Key, Attr

            dynamodb = boto3.resource('dynamodb', region\_name='us-east-1')

            table = dynamodb.Table('iotdata')

            now = datetime.datetime.now()

            startdate = now.strftime("%Y-%m")

            response = table.query(

                KeyConditionExpression=Key('deviceid').eq('deviceid\_dorachua')

                                      & Key('datetimeid').begins\_with(startdate),

                ScanIndexForward=False

            )

            items = response['Items']

            n=10 # limit to last 10 items

            data = items[:n]

            data\_reversed = data[::-1]

            return data\_reversed

    except:

        import sys

        print(sys.exc\_info()[0])

        print(sys.exc\_info()[1])

if \_\_name\_\_ == "\_\_main\_\_":

    get\_data\_from\_dynamodb()

### detect\_motion.py

from gpiozero import MotionSensor,Buzzer

import time

from gpiozero import LED

from picamera import PiCamera

import boto3

import botocore

import json

pir = MotionSensor(12, sample\_rate=5,queue\_len=1)

#bz = Buzzer(13)

camera = PiCamera()

# Set the filename and bucket name

bucket = 'sp-p1828261-s3-bucket' # replace with your own unique bucket name

exists = True

full\_path = '/home/pi/Desktop/CA2/image1.jpg'

file\_name = 'image1.jpg'

def uploadToS3():

    s3 = boto3.resource('s3')

    bucket = 'sp-p1828261-s3-bucket' # replace with your own unique bucket name

    exists = True

    try:

        s3.meta.client.head\_bucket(Bucket=bucket)

    except botocore.exceptions.ClientError as e:

        error\_code = int(e.response['Error']['Code'])

        if error\_code == 404:

            exists = False

    if exists == False:

        s3.create\_bucket(Bucket=bucket,CreateBucketConfiguration={

        'LocationConstraint': 'us-east-1'})

    camera.capture(full\_path)

    s3.Object(bucket, file\_name).put(Body=open(full\_path, 'rb'))

    print("File Uploaded")

def downloadFromS3(timestring):

    s3 = boto3.resource('s3')

    bucket = 'sp-p1828261-s3-bucket' # replace with your own unique bucket name

    exists = True

    try:

        s3.meta.client.head\_bucket(Bucket=bucket)

    except botocore.exceptions.ClientError as e:

        error\_code = int(e.response['Error']['Code'])

        if error\_code == 404:

            exists = False

    if exists == False:

        s3.create\_bucket(Bucket=bucket,CreateBucketConfiguration={

        'LocationConstraint': 'us-east-1'})

    camera.capture(full\_path)

    s3.Object(bucket, file\_name).download\_file(timestring+".jpg")

    print("File Uploaded")

# Minimum time between captures

DELAY = 5

from twilio.rest import Client

account\_sid = "AC5974189b13df2f64fd39aadd3c3aeb98"

auth\_token = "3bcf7fd3659f7354636ccace20321488"

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

my\_hp = "+6581869959"

twilio\_hp = "+13345083131"

sms = "There seems to be someone in your room currently, "

def detect\_labels(bucket, key, max\_labels=10, min\_confidence=90, region="us-east-1"):

    rekognition = boto3.client("rekognition", region)

    response = rekognition.detect\_labels(

        Image={

            "S3Object": {

                "Bucket": bucket,

                "Name": key,

            }

        },

        MaxLabels=max\_labels,

        MinConfidence=min\_confidence,

    )

    return response['Labels']

def detect\_faces(bucket, key, max\_labels=10, min\_confidence=90, region="us-east-1"):

    rekognition = boto3.client("rekognition", region)

    response = rekognition.detect\_faces(

        Image={

            "S3Object": {

                "Bucket": bucket,

                "Name": key,

            }

        },

        Attributes=['ALL']

    )

    return response['FaceDetails']

while True:

    #bz.off()

    pir.wait\_for\_motion()

    # Take photo and send to s3, then send to aws rekognition

    # If human is detected, send sms through twilio

    uploadToS3()

    highestconfidence = 0

    best\_bet\_item = "Unknown"

    for label in detect\_labels(bucket, file\_name):

        print("{Name} - {Confidence}%".format(\*\*label))

        if label["Name"] == "Human" or label["Name"] == "Person":

            print("human")

            for faceDetail in detect\_faces(bucket, file\_name):

                ageLow = faceDetail['AgeRange']['Low']

                ageHigh = faceDetail['AgeRange']['High']

                ageinfo = 'Age between {} and {} years old'.format(ageLow,ageHigh)

            to\_send = sms + ageinfo + ". Time: " + time.strftime("%Y-%m-%dT%H:%M:%S", time.gmtime())

            message = client.api.account.messages.create(to=my\_hp,from\_=twilio\_hp,body=to\_send)

            timestring = time.strftime("%Y-%m-%dT%H:%M:%S", time.gmtime())

            #bz.on()

            downloadFromS3(timestring)

            print ("Motion Detected " +timestring)

            break

        """

        if label["Confidence"] >= highestconfidence:

            highestconfidence = label["Confidence"]

            best\_bet\_item = label["Name"]

        """

    if best\_bet\_item!= "Unknown":

        print("This should be a {} with confidence {}".format(best\_bet\_item, highestconfidence))

    pir.wait\_for\_no\_motion()

    #bz.off()

    time.sleep(DELAY)

### index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>SB Admin 2 - Dashboard</title>

    <!-- Custom fonts for this template-->

    <link href="static/vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

    <link

        href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i"

        rel="stylesheet">

    <!-- Custom styles for this template-->

    <link href="static/css/sb-admin-2.min.css" rel="stylesheet">

    <style> #chartDiv {width:100%;}</style>

    <title>Google Charts with Flask</title>

    <script type="text/javascript" src="https://code.jquery.com/jquery-3.2.1.js"></script>

    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

    <script type="text/javascript">

        google.charts.load('current', {'packages':['corechart','table']});

       // Set a callback to run when the Google Visualization API is loaded.

       google.charts.setOnLoadCallback(googlecharts\_is\_ready);

       var chart;

       var graphdata;

       function reset\_status\_messages(){

           $("#status").html("")

       }

       function googlecharts\_is\_ready(){

       }

       function getNewData(){

           $("#status").html("Fetching data to plot graph...");

           jQuery.ajax({

               url: "/api/getdata" ,

               type: 'POST',

               error: function(jqXHR,textStatus, errorThrown ){

                    console.log("Error while ajax:" + textStatus)

               },

               success: function(ndata, textStatus, xhr){

                   //console.log(ndata)

                   //console.log(ndata.chart\_data)

                   $("#status").html("Data fetched! Now plotting graph!");

                   chartdata = ndata.chart\_data

                   graphdata = createDataTable(chartdata)

                   drawLineChart(graphdata)

                   drawDataTable(graphdata)

                   $("#status").html("Graph plotted");

               }//end success

           });//end ajax

         } //end getNewData

       function createDataTable(newdata){

           graphdata = new google.visualization.DataTable();

           graphdata.addColumn('string', 'Time');

           graphdata.addColumn('number', 'Light');

           var newdata = JSON.parse(newdata);

            for (index=0;index<newdata.length;index++){

                datetime = (newdata[index].datetimeid)

                datetime = datetime.substring(0, 19) //+ "+0000"

                jsdatetime = new Date(Date.parse(datetime));

                jstime = jsdatetime.toLocaleTimeString();

                light = parseInt(newdata[index].value);

                graphdata.addRows([[jstime,light]]);

            }//end for

            return graphdata

        }

        function drawDataTable(graphdata){

            var table = new google.visualization.Table(document.getElementById('table\_div'));

            table.draw(graphdata, {showRowNumber: true, width: '100%', height: '100%'});

        }//end drawTable

        function drawLineChart(graphdata) {

            chart = new google.visualization.LineChart(

            document.getElementById('chart\_div'));

            chart.draw(graphdata, {legend: 'none', vAxis: {baseline: 0},

                colors: ['#A0D100']});

            return

        } //end drawChart

        $(document).ready(function(){

            reset\_status\_messages()

            setInterval(function () {

                getNewData()

            }, 10000);

        });

</script>

<script>

    function turnon(){

      $.ajax({url: "writeLED/On",

              success: function(result){

                            $("#ledstatus").html(result);

                            }

            })

    }

    function turnoff(){

      $.ajax({url: "writeLED/Off",

              success: function(result){

                            $("#ledstatus").html(result);

                            }

      })

    }

    $(document).ready(function(){

        $("#b1").click(function(){

            turnon();

              });

            $("#b2").click(function(){

                     turnoff();

              });

         });

      </script>

<script>

    function motionon(){

      $.ajax({url: "motion/Start",

              success: function(result){

                            $("#motionstatus").html(result);

                            }

            })

    }

    function motionoff(){

      $.ajax({url: "motion/Stop",

              success: function(result){

                            $("#motionstatus").html(result);

                            }

      })

    }

    $(document).ready(function(){

        $("#m1").click(function(){

            motionon();

              });

            $("#m2").click(function(){

                     motionoff();

              });

         });

      </script>

<script>

    function getDHT(){

          jQuery.ajax({

               url: "/getDHT" ,

               type: 'POST',

               success: function(ndata, textStatus, xhr){

                   console.log(ndata)

                   $("#dhtstatus").html("Temperature:" + ndata.temperature + "<br/>" + "Humidity: " + ndata.humidity);

               }//end success

           });//end ajax

         } //end

         $(document).ready(function(){

           setInterval(function () {

               getDHT();

           }, 5000);

       });

 </script>

</head>

<body id="page-top">

    <!-- Page Wrapper -->

    <div id="wrapper">

        <!-- Sidebar -->

        <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

            <!-- Sidebar - Brand -->

            <a class="sidebar-brand d-flex align-items-center justify-content-center" href="index.html">

                <div class="sidebar-brand-icon rotate-n-15">

                    <i class="fas fa-laugh-wink"></i>

                </div>

                <div class="sidebar-brand-text mx-3">SB Admin <sup>2</sup></div>

            </a>

            <!-- Divider -->

            <hr class="sidebar-divider my-0">

            <!-- Nav Item - Dashboard -->

            <li class="nav-item active">

                <a class="nav-link" href="index.html">

                    <i class="fas fa-fw fa-tachometer-alt"></i>

                    <span>Dashboard</span></a>

            </li>

            <!-- Heading -->

            <div class="sidebar-heading"></div>

            <!-- Nav Item - Pages Collapse Menu -->

            <li class="nav-item">

            </li>

            <!-- Nav Item - Utilities Collapse Menu -->

            <li class="nav-item">

            </li>

            <!-- Heading -->

            <div class="sidebar-heading">

            </div>

            <!-- Nav Item - Pages Collapse Menu -->

            <li class="nav-item">

            </li>

            <!-- Nav Item - Charts -->

            <li class="nav-item">

            </li>

            <!-- Nav Item - Tables -->

            <li class="nav-item">

            </li>

        </ul>

        <!-- End of Sidebar -->

        <!-- Content Wrapper -->

        <div id="content-wrapper" class="d-flex flex-column">

            <!-- Main Content -->

            <div id="content">

                <!-- Topbar -->

                <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-top shadow">

                    <!-- Sidebar Toggle (Topbar) -->

                    <button id="sidebarToggleTop" class="btn btn-link d-md-none rounded-circle mr-3">

                        <i class="fa fa-bars"></i>

                    </button>

                    <!-- Topbar Search -->

                    <form

                        class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2 my-md-0 mw-100 navbar-search">

                        <div class="input-group">

                            <input type="text" class="form-control bg-light border-0 small" placeholder="Search for..."

                                aria-label="Search" aria-describedby="basic-addon2">

                            <div class="input-group-append">

                                <button class="btn btn-primary" type="button">

                                    <i class="fas fa-search fa-sm"></i>

                                </button>

                            </div>

                        </div>

                    </form>

                    <!-- Topbar Navbar -->

                    <ul class="navbar-nav ml-auto">

                        <!-- Nav Item - Search Dropdown (Visible Only XS) -->

                        <li class="nav-item dropdown no-arrow d-sm-none">

                            <a class="nav-link dropdown-toggle" href="#" id="searchDropdown" role="button"

                                data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

                                <i class="fas fa-search fa-fw"></i>

                            </a>

                            <!-- Dropdown - Messages -->

                            <div class="dropdown-menu dropdown-menu-right p-3 shadow animated--grow-in"

                                aria-labelledby="searchDropdown">

                                <form class="form-inline mr-auto w-100 navbar-search">

                                    <div class="input-group">

                                        <input type="text" class="form-control bg-light border-0 small"

                                            placeholder="Search for..." aria-label="Search"

                                            aria-describedby="basic-addon2">

                                        <div class="input-group-append">

                                            <button class="btn btn-primary" type="button">

                                                <i class="fas fa-search fa-sm"></i>

                                            </button>

                                        </div>

                                    </div>

                                </form>

                            </div>

                        </li>

                        <!-- Nav Item - Alerts -->

                        <li class="nav-item dropdown no-arrow mx-1">

                            <a class="nav-link dropdown-toggle" href="#" id="alertsDropdown" role="button"

                                data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

                                <i class="fas fa-bell fa-fw"></i>

                                <!-- Counter - Alerts -->

                                <span class="badge badge-danger badge-counter"></span>

                            </a>

                            <!-- Dropdown - Alerts -->

                            <div class="dropdown-list dropdown-menu dropdown-menu-right shadow animated--grow-in"

                                aria-labelledby="alertsDropdown">

                                <h6 class="dropdown-header">

                                    Alerts Center

                                </h6>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="mr-3">

                                        <div class="icon-circle bg-primary">

                                            <i class="fas fa-file-alt text-white"></i>

                                        </div>

                                    </div>

                                    <div>

                                    </div>

                                </a>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="mr-3">

                                        <div class="icon-circle bg-success">

                                            <i class="fas fa-donate text-white"></i>

                                        </div>

                                    </div>

                                    <div>

                                </a>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="mr-3">

                                        <div class="icon-circle bg-warning">

                                            <i class="fas fa-exclamation-triangle text-white"></i>

                                        </div>

                                    </div>

                                    <div>

                                </a>

                                <a class="dropdown-item text-center small text-gray-500" href="#">Show All Alerts</a>

                            </div>

                        </li>

                        <!-- Nav Item - Messages -->

                        <li class="nav-item dropdown no-arrow mx-1">

                            <a class="nav-link dropdown-toggle" href="#" id="messagesDropdown" role="button"

                                data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

                                <i class="fas fa-envelope fa-fw"></i>

                                <!-- Counter - Messages -->

                                <span class="badge badge-danger badge-counter">7</span>

                            </a>

                            <!-- Dropdown - Messages -->

                            <div class="dropdown-list dropdown-menu dropdown-menu-right shadow animated--grow-in"

                                aria-labelledby="messagesDropdown">

                                <h6 class="dropdown-header">

                                    Message Center

                                </h6>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="dropdown-list-image mr-3">

                                        <div class="status-indicator bg-success"></div>

                                    </div>

                                    <div class="font-weight-bold">

                                    </div>

                                </a>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="dropdown-list-image mr-3">

                                        <div class="status-indicator"></div>

                                    </div>

                                    <div>

                                    </div>

                                </a>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="dropdown-list-image mr-3">

                                        <div class="status-indicator bg-warning"></div>

                                    </div>

                                    <div>

                                    </div>

                                </a>

                                <a class="dropdown-item d-flex align-items-center" href="#">

                                    <div class="dropdown-list-image mr-3">

                                        <div class="status-indicator bg-success"></div>

                                    </div>

                                    <div>

                                    </div>

                                </a>

                                <a class="dropdown-item text-center small text-gray-500" href="#"></a>

                            </div>

                        </li>

                    </ul>

                </nav>

                <!-- End of Topbar -->

                <!-- Begin Page Content -->

                <div class="container-fluid"></div>

                    <!-- Page Heading -->

                    <div class="d-sm-flex align-items-center justify-content-between mb-4">

                        <h1 class="h3 mb-0 text-gray-800">Dashboard</h1>

                    </div>

                    <!-- Content Row -->

                    <div class="row">

                        <!-- Earnings (Monthly) Card Example -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-success shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-success text-uppercase mb-1">

                                                Current Temperature and Humidity</div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800"><span id="dhtstatus"></span></div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-dollar-sign fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- Earnings (Monthly) Card Example -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-info shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-info text-uppercase mb-1">LED Controls</div>

                                            <div class="row no-gutters align-items-center">

                                                <button id="b1">Turn on</button>

                                                <button id="b2">Turn off</button>

                                                <h2 id="ledstatus"></h2>

                                            </div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-clipboard-list fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- Pending Requests Card Example -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-warning shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-warning text-uppercase mb-1">

                                                Motion Sensor Controls</div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800">

                                                <button id="m1">Turn on</button>

                                                <button id="m2">Turn off</button>

                                                <h2 id="motionstatus"></h2></div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-comments fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <!-- Content Row -->

                    <div class="row">

                        <!-- Area Chart -->

                        <div class="col-xl-8 col-lg-7">

                            <div class="card shadow mb-4">

                                <!-- Card Header - Dropdown -->

                                <div

                                    class="card-header py-3 d-flex flex-row align-items-center justify-content-between">

                                    <h6 class="m-0 font-weight-bold text-primary">Light Level Graph</h6>

                                    <div id="chart\_div" style="width:100%"></div>

                                </div>

                                <!-- Card Body -->

                            </div>

                        </div>

                        <!-- Area Chart -->

                        <div class="col-xl-8 col-lg-7">

                            <div class="card shadow mb-4">

                                <!-- Card Header - Dropdown -->

                                <div

                                    class="card-header py-3 d-flex flex-row align-items-center justify-content-between">

                                    <h6 class="m-0 font-weight-bold text-primary">Light Level Table</h6>

                                    <div id="table\_div" style="width:100%"></div>

                                </div>

                                <!-- Card Body -->

                            </div>

                        </div>

            </div>

            <!-- End of Main Content -->

            <!-- Footer -->

            <footer class="sticky-footer bg-white">

                <div class="container my-auto">

                    <div class="copyright text-center my-auto">

                        <span>Copyright &copy; Your Website 2020</span>

                    </div>

                </div>

            </footer>

            <!-- End of Footer -->

        </div>

        <!-- End of Content Wrapper -->

    </div>

    <!-- End of Page Wrapper -->

    <!-- Scroll to Top Button-->

    <a class="scroll-to-top rounded" href="#page-top">

        <i class="fas fa-angle-up"></i>

    </a>

    <!-- Logout Modal-->

    <div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"

        aria-hidden="true">

        <div class="modal-dialog" role="document">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>

                    <button class="close" type="button" data-dismiss="modal" aria-label="Close">

                        <span aria-hidden="true">×</span>

                    </button>

                </div>

                <div class="modal-body">Select "Logout" below if you are ready to end your current session.</div>

                <div class="modal-footer">

                    <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancel</button>

                    <a class="btn btn-primary" href="login.html">Logout</a>

                </div>

            </div>

        </div>

    </div>

    <!-- Bootstrap core JavaScript-->

    <script src="vendor/jquery/jquery.min.js"></script>

    <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

    <!-- Core plugin JavaScript-->

    <script src="vendor/jquery-easing/jquery.easing.min.js"></script>

    <!-- Custom scripts for all pages-->

    <script src="js/sb-admin-2.min.js"></script>

    <!-- Page level plugins -->

    <script src="vendor/chart.js/Chart.min.js"></script>

    <!-- Page level custom scripts -->

    <script src="js/demo/chart-area-demo.js"></script>

    <script src="js/demo/chart-pie-demo.js"></script>

</body>

</html>

# Section 5 Task List

A table listing members names and the parts of the assignment they worked on

|  |  |  |
| --- | --- | --- |
| Name of member | Part of project worked on | Contribution percentage |
| Vincent Sow Zhao Bin | LED Control  Sending LDR values using MQTT  Storing LDR values in DynamoDB  Retrieve DLR Values from DynamoDB, and display as graph in Flask Server | 50% |
| Chee Poh Chuan | DHT-11 real-time values  Alert system using motion sensor, pi camera.  Upload and download photo using S3 and detecting human using AWS rekognition  Send SMS to alert user of unwanted presence | 50% |

# Section 6 References

<https://docs.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax>

<https://stackoverflow.com/questions/63342520/raspberry-pi-error-with-picamera-exc-picamerammalerror-failed-to-enable-connect>

<https://stackoverflow.com/questions/6809402/python-maximum-recursion-depth-exceeded-while-calling-a-python-object>

<https://stackabuse.com/file-management-with-aws-s3-python-and-flask/>

**-- End of CA2 Step-by-step tutorial --**