**SCHOOL OF COMPUTING (SOC)**

|  |  |
| --- | --- |
| **Date of Submission:** | 21/3/21 |

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

|  |  |
| --- | --- |
| **Class:** | DISM/FT/3A/42 |

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

|  |  |
| --- | --- |
| **Student ID**  1804205 | **Name**  Thien Jun Heng |
| 1804148 | Boobaesh Kumaran |

**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 2](#_Toc61178270)

[A. Where we have uploaded our tutorial 2](#_Toc61178271)

[B. What is the application about? 2](#_Toc61178272)

[C. How does the final RPI set-up looks like? 2](#_Toc61178273)

[D. How does the web or mobile application look like? 2](#_Toc61178274)

[E. System architecture of our system 3](#_Toc61178275)

[F. Evidence that we have met basic requirements 3](#_Toc61178276)

[G. Bonus features on top of basic requirements 4](#_Toc61178277)

[A. Quick-start guide (Readme first) 4](#_Toc61178278)

[Section 2 Hardware requirements 5](#_Toc61178279)

[Hardware checklist 5](#_Toc61178280)

[Hardware setup instructions 5](#_Toc61178281)

[Fritzing Diagram 5](#_Toc61178282)

[Section 3 Software Requirements 6](#_Toc61178283)

[Software checklist 6](#_Toc61178284)

[Software setup instructions 6](#_Toc61178285)

[Section 4 Source codes 6](#_Toc61178286)

[server.py 6](#_Toc61178287)

[index.html 6](#_Toc61178288)

[Section 5 Task List 7](#_Toc61178289)

[Section 6 Any other section you want to add 7](#_Toc61178290)

[Section 7 References 7](#_Toc61178291)

# 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>

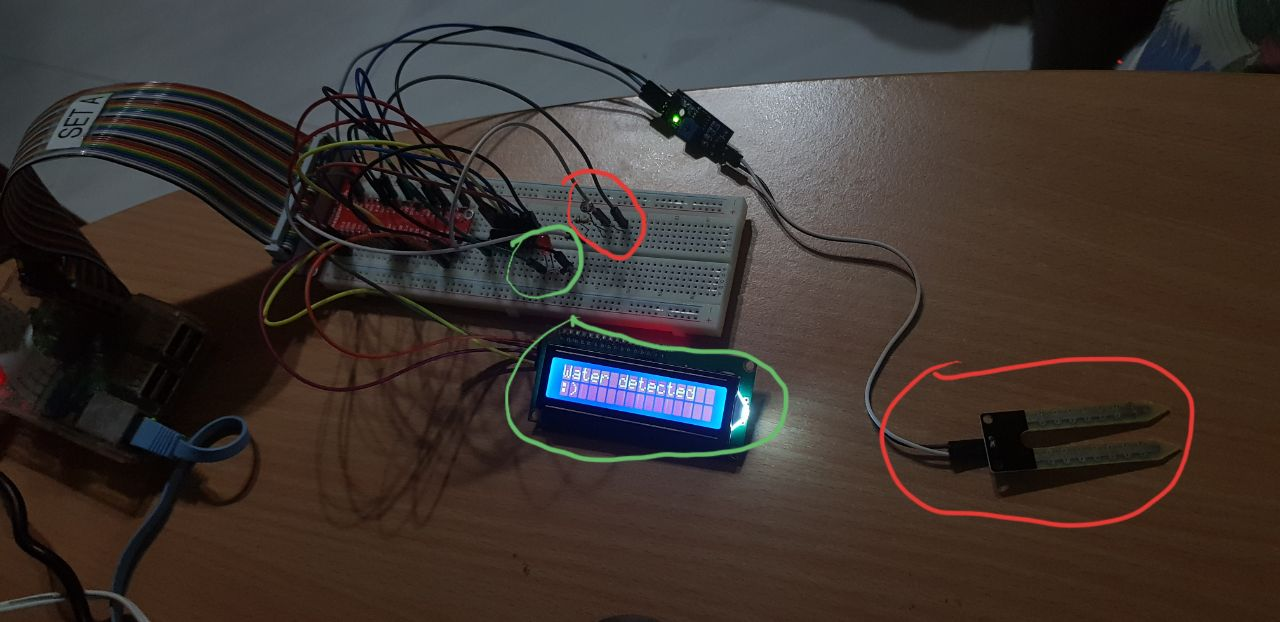
|  |  |
| --- | --- |
| **Youtube** | https://youtu.be/Ve8tyvGSvNQ |
| **Public tutorial link** |  |

* 1. What is the application about?

This project is a prototype IoT device that looks after plants. It comes with a LCD screen that reads whether the soil has detected water or not. The device also comes with a light sensor that detects whether the plant has adequate sunlight. When the surroundings go dark, the user would get a text message asking to bring the plant to a place that’s brighter for it to grow.

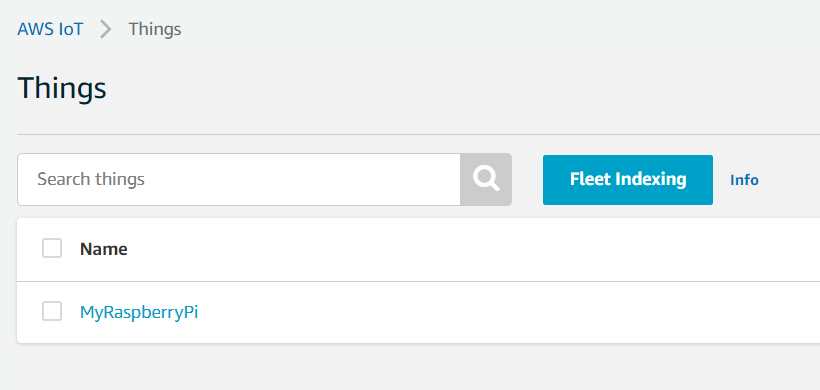
* 1. How does the final RPI set-up looks like?

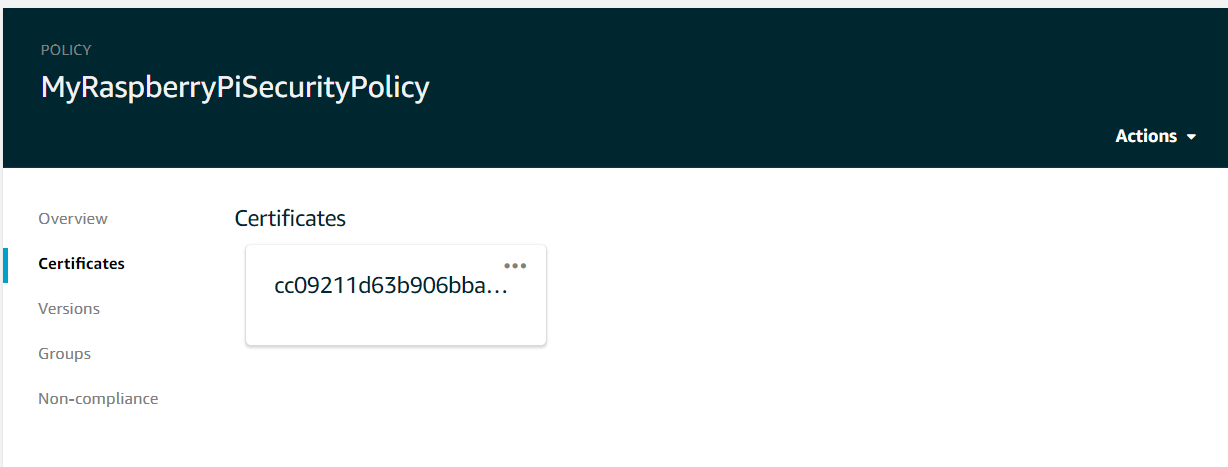
Provide a photo of your final RPI hardware set-up. You may want to mark-up (annotate or draw arrows) and refer to this in Section F for instance.

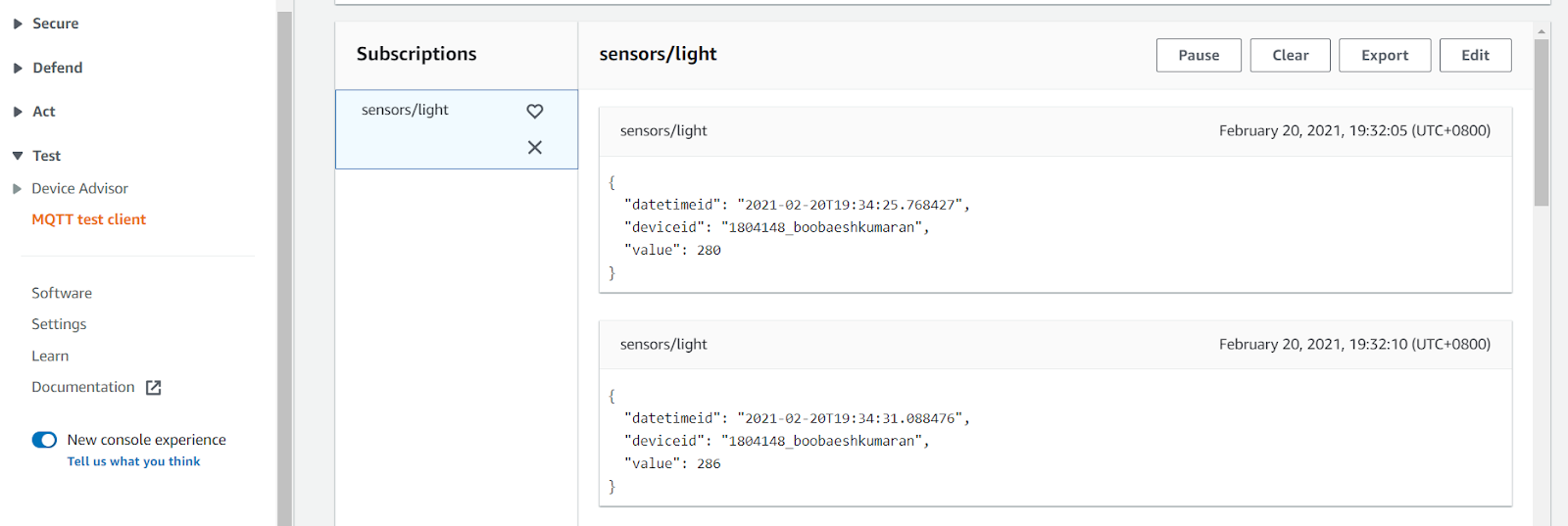


* 1. How does the web or mobile application look like?

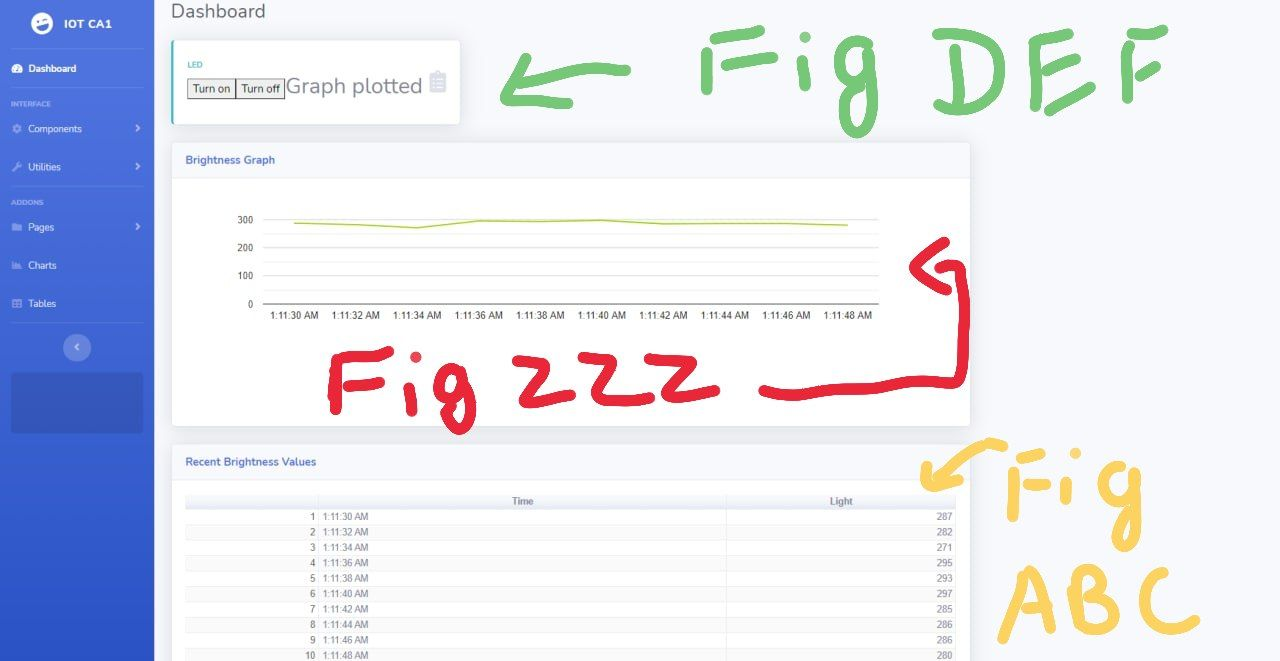
Provide at least one screenshot of your web app, and more if your web app consists of more than 1 page. Otherwise, I will assume your webapp only can show 1 page. Label your screenshots so that they may be referenced in Section F.







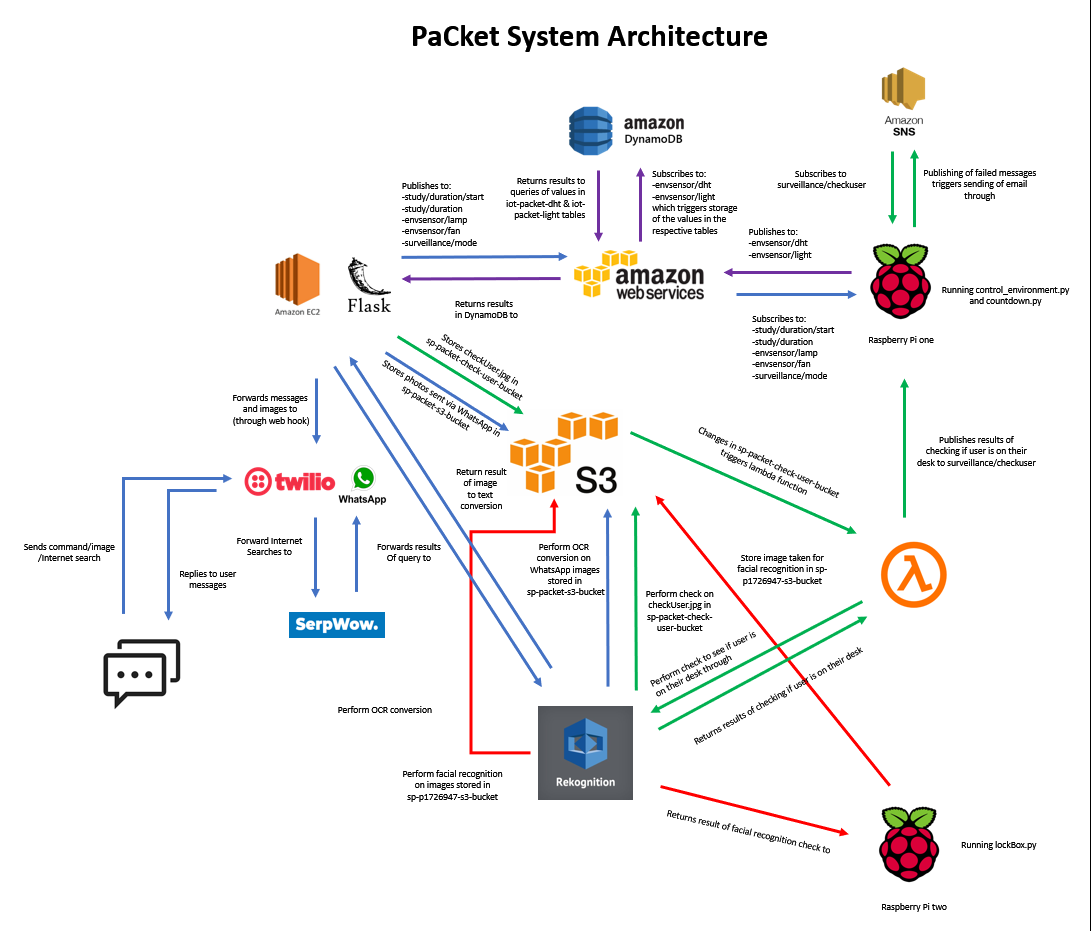




Refer to the labels in Section F

* 1. System architecture of our system

Provide a hand-drawn or computer-drawn system architecture diagram please. Example given below.



* 1. Evidence that we have met basic requirements

Provide bullet list to describe how your group has met basic requirements

|  |  |
| --- | --- |
| Requirement | Evidence |
| Used three sensors | Marked in **green** are actuators (LCD Screen and LED)  Marked in **red** are sensors (Soil Moisture Sensor and Light Sensor) |
| Used MQTT | Our MQTT endpoint --> sensors/light  Example of data sent through MQTT : |
| Stored data in cloud | Stored data in DynamoDB |
| Used cloud service | Use AWS Rekognition, used DynamoDB, used mqtt, used S3 bucket container to upload images |
| Provide real-time sensor value / status | Refer to screenshot labelled Fig ZZZ |
| Provide historical sensor value/ status | Refer to screenshot labelled Fig ZZZ |
| Control actuator | Refer to screenshot labelled Fig DEF |

* 1. Bonus features on top of basic requirements

Provide bullet list of the bonus features you have added on top of basic requirements

1. SMS to alert user of whether there is light for the plant to grow well and healthily
   1. Quick-start guide (Readme first)

Give a few lines of basic instructions on how I need to run your app, e.g

1. First connect hardware as in Section 1
2. Then run aws\_pubsub\_edited.py file to publish the light values to subscriber
3. Then run dynamodb.py to query the data from querying the DB
4. Then run boto\_s3\_1.py to capture images and uploads to AWS S3 container
5. Then run mcp3008\_light\_twilio.py to sends light values to your smartphone via SMS

# Section 2 Hardware requirements

Hardware checklist

1. 1 i2C 12x6 LCD Screen
2. 1 Soil Moisture Detector
3. 1 LED
4. 1 LDR
5. MCP3008
6. 330 Ohm resistor
7. 10K Ohm resistor

Hardware setup instructions

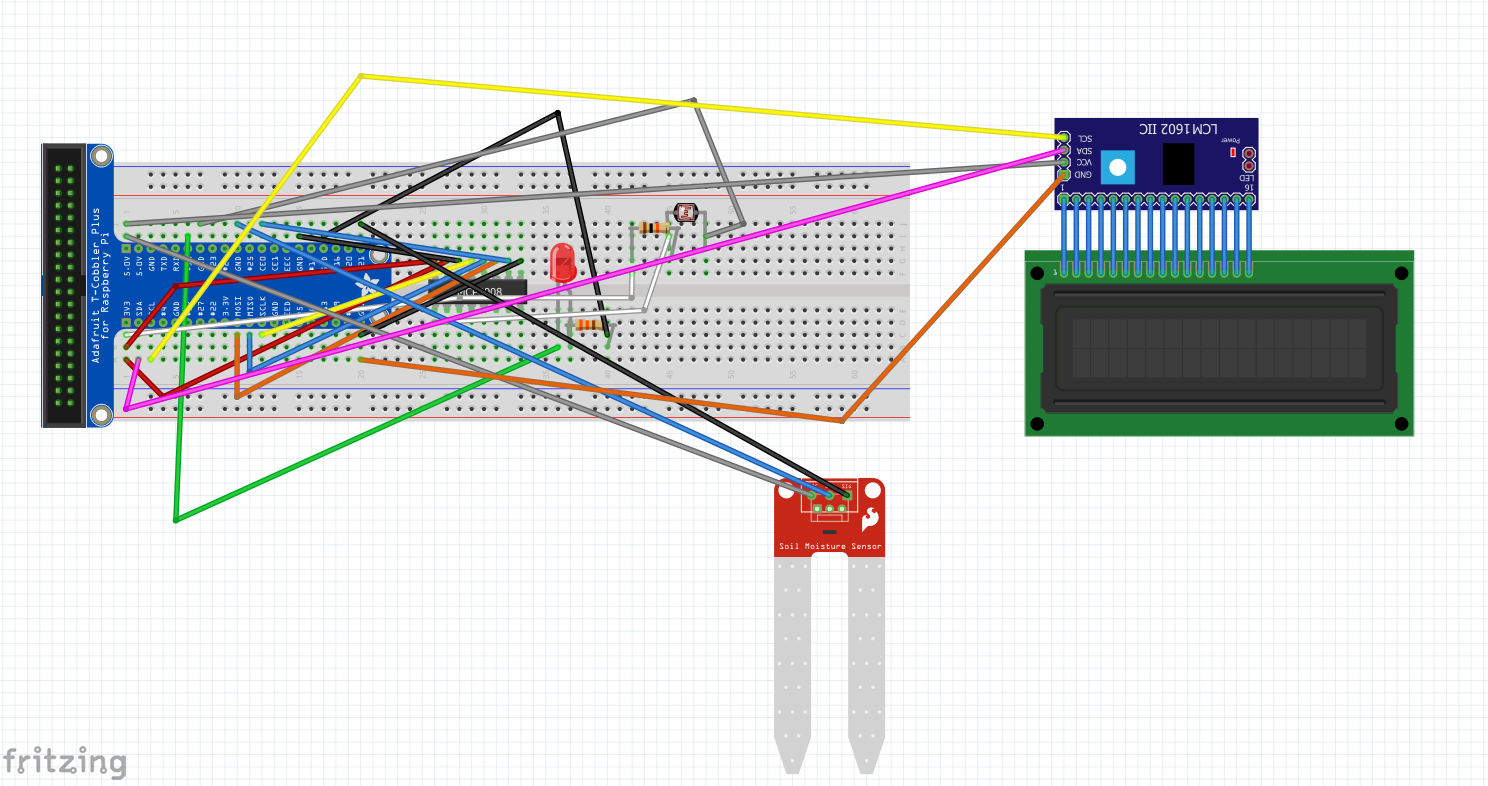
There are many types of soil moisture sensors that are out there. So please understand that connections for certain sensors may vary



For the case of my assignment, I have used this particular soil moisture detector

Fritzing Diagram

Paste a Fritzing diagram of your setup here



You can get the Fritzing software at Blackboard Labs folder (third link from top)

# Section 3 Software Requirements

Software checklist

There is no additional Python or other libraries

Software setup instructions

There is no software setup instructions

# Section 4 Source codes

All source codes, including Python, HTML files etc

### server.py

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

from rpi\_lcd import LCD

import mysql.connector

import sys

import time

from time import sleep

import RPi.GPIO as GPIO

import json

import numpy

import datetime

import decimal

from gpiozero import LED

import dynamodb

import jsonconverter as jsonc

import gevent

import gevent.monkey

from gevent.pywsgi import WSGIServer

gevent.monkey.patch\_all()

led = LED(18) #pin 18 for LED

lcd = LCD()

channel = 21 #GPIO pin 21 for soil moisture sensor

GPIO.setmode(GPIO.BCM)

GPIO.setup(channel, GPIO.IN)

def callback(channel):

   if GPIO.input(channel):

      lcd.text('Water is not', 1)

      lcd.text('detected :(', 2)

   else:

      lcd.text('Water detected', 1)

      lcd.text(':)', 2)

   time.sleep(1)

GPIO.add\_event\_detect(channel, GPIO.BOTH, bouncetime=300) # lets us know when the pin goes HIGH or LOW

GPIO.add\_event\_callback(channel, callback) # assigns function to GPIO pin, Runs function on change

def ledOn():

  led.on()

  return "LED has been turned on"

def ledOff():

  led.off()

  return "LED has been turned off"

def ledStatus():

  if led.is\_lit:

     return 'On'

  else:

    return 'Off'

app = Flask(\_\_name\_\_)

@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 index():

    return render\_template('index.html')

@app.route("/readLED/")

def readPin():

   response = ledStatus()

   templateData = {

      'title' : 'Status of LED: ',

      'response' : response

   }

   return render\_template('pin.html', \*\*templateData)

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

def writePin(status):

   if status == 'On':

     response = ledOn()

   else:

     response = ledOff()

   templateData = {

      'title' : 'Status of LED',

      'response' : response

   }

   return render\_template('pin.html', \*\*templateData)

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])

### 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

            from boto3.dynamodb.conditions import Key, Attr

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

            table = dynamodb.Table('iotdata')

            startdate = '2021-02' #ensure this value gets changed every month/year

            response = table.query(

                KeyConditionExpression=Key('deviceid').eq('1804148\_boobaeshkumaran')

                                      & 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()

### boto\_s3\_1.py

import boto3

import botocore

from time import sleep

# Create an S3 resource

s3 = boto3.resource('s3')

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

file\_name = 'image1.jpg'

def takePhotoWithPiCam():

    from picamera import PiCamera

    camera = PiCamera()

    sleep(5)

    camera.capture(full\_path)

    sleep(3)

# Set the filename and bucket name

bucket = 'sp-p1804148-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'})

# Take a photo

takePhotoWithPiCam()

# Upload a new file

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

print("File uploaded")

### aws\_pubsub\_edited.py

# Import SDK packages

from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

from time import sleep

from gpiozero import MCP3008

mcp3008 = 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 = "a2vrpf387xyqnk-ats.iot.us-east-1.amazonaws.com"

rootCAPath = "rootca.pem"

certificatePath = "certificate.pem.crt"

privateKeyPath = "private.pem.key"

my\_rpi = AWSIoTMQTTClient("p1804148basicPubSub")

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:

    import random

    #light = random.randint(1,1024)

    light = (1024\*(1.0-mcp3008.value))

    light = round(light)

    loopCount = loopCount+1

    message = {}

    message["deviceid"] = "1804148\_boobaeshkumaran"

    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)

### 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>IoT CA1</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()

            }, 3000);

        });

</script>

<script>

    function turnon(){

      $.ajax({url: "writeLED/On",

              success: function(result){

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

                            }

            })

    }

    function turnoff(){

      $.ajax({url: "writeLED/Off",

              success: function(result){

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

                            }

      })

    }

    $(document).ready(function(){

        $("#b1").click(function(){

               turnon();

        });

      $("#b2").click(function(){

               turnoff();

        });

   });

</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">IoT CA1</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>

            <!-- Divider -->

            <hr class="sidebar-divider">

            <!-- Heading -->

            <div class="sidebar-heading">

                Interface

            </div>

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

            <li class="nav-item">

                <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseTwo"

                    aria-expanded="true" aria-controls="collapseTwo">

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

                    <span>Components</span>

                </a>

                <div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

                    <div class="bg-white py-2 collapse-inner rounded">

                        <h6 class="collapse-header">Custom Components:</h6>

                        <a class="collapse-item" href="buttons.html">Buttons</a>

                        <a class="collapse-item" href="cards.html">Cards</a>

                    </div>

                </div>

            </li>

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

            <li class="nav-item">

                <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseUtilities"

                    aria-expanded="true" aria-controls="collapseUtilities">

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

                    <span>Utilities</span>

                </a>

                <div id="collapseUtilities" class="collapse" aria-labelledby="headingUtilities"

                    data-parent="#accordionSidebar">

                    <div class="bg-white py-2 collapse-inner rounded">

                        <h6 class="collapse-header">Custom Utilities:</h6>

                        <a class="collapse-item" href="utilities-color.html">Colors</a>

                        <a class="collapse-item" href="utilities-border.html">Borders</a>

                        <a class="collapse-item" href="utilities-animation.html">Animations</a>

                        <a class="collapse-item" href="utilities-other.html">Other</a>

                    </div>

                </div>

            </li>

            <!-- Divider -->

            <hr class="sidebar-divider">

            <!-- Heading -->

            <div class="sidebar-heading">

                Addons

            </div>

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

            <li class="nav-item">

                <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapsePages"

                    aria-expanded="true" aria-controls="collapsePages">

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

                    <span>Pages</span>

                </a>

                <div id="collapsePages" class="collapse" aria-labelledby="headingPages" data-parent="#accordionSidebar">

                    <div class="bg-white py-2 collapse-inner rounded">

                        <h6 class="collapse-header">Login Screens:</h6>

                        <a class="collapse-item" href="login.html">Login</a>

                        <a class="collapse-item" href="register.html">Register</a>

                        <a class="collapse-item" href="forgot-password.html">Forgot Password</a>

                        <div class="collapse-divider"></div>

                        <h6 class="collapse-header">Other Pages:</h6>

                        <a class="collapse-item" href="404.html">404 Page</a>

                        <a class="collapse-item" href="blank.html">Blank Page</a>

                    </div>

                </div>

            </li>

            <!-- Nav Item - Charts -->

            <li class="nav-item">

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

                    <i class="fas fa-fw fa-chart-area"></i>

                    <span>Charts</span></a>

            </li>

            <!-- Nav Item - Tables -->

            <li class="nav-item">

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

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

                    <span>Tables</span></a>

            </li>

            <!-- Divider -->

            <hr class="sidebar-divider d-none d-md-block">

            <!-- Sidebar Toggler (Sidebar) -->

            <div class="text-center d-none d-md-inline">

                <button class="rounded-circle border-0" id="sidebarToggle"></button>

            </div>

            <!-- Sidebar Message -->

            <div class="sidebar-card">

                <img class="sidebar-card-illustration mb-2" src="img/undraw\_rocket.svg" alt="">

            </div>

        </ul>

        <!-- End of Sidebar -->

        <!-- Content Wrapper -->

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

            <!-- Main Content -->

            <div id="content">

                <!-- Begin Page Content -->

                <div class="container-fluid">

                    <!-- 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>

                        <a href="#" class="d-none d-sm-inline-block btn btn-sm btn-primary shadow-sm"><i

                                class="fas fa-download fa-sm text-white-50"></i> Generate Report</a>

                    </div>

                    <!-- Content Row -->

                    <div class="row">

                        <!-- LED Control -->

                        <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

                                            </div>

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

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

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

                                                <h2 id="status"></h2>

                                            </div>

                                        </div>

                                        <div class="col-auto">

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

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <!-- Content Row -->

                    <div class="row">

                        <input id="buttonloadchart" type="button" onclick="loadChart()" value="Update graph">

                        <div id="status"></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">Brightness Graph</h6>

                                </div>

                                <!-- Card Body -->

                                <div class="card-body">

                                    <div class="chart-area">

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

                                    </div>

                                </div>

                            </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">Recent Brightness Values</h6>

                                </div>

                                <!-- Card Body -->

                                <div class="card-body">

                                    <div class="chart-area">

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

                                    </div>

                                </div>

                            </div>

                        </div>

                    <!-- Content Row -->

                    <div class="row">

                        <!-- Content Column -->

                        <div class="col-lg-6 mb-4">

                            </div>

                            <!-- Color System -->

                            <div class="row">

                                <div class="col-lg-6 mb-4">

                                </div>

                                <div class="col-lg-6 mb-4">

                                </div>

                            </div>

                        </div>

                        <div class="col-lg-6 mb-4">

                        </div>

                    </div>

                </div>

                <!-- /.container-fluid -->

            </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; Boobaesh Kumaran DISM/FT/3A/42</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  Thien Jun Heng  Boobaesh Kumaran | Part of project worked on  Connecting the setup  Filming the video  Debugging of codes  Did the step-by-step tutorial  Publishing the tutorial online  Coding the project  Posting the video to youtube | Contribution percentage  35%  65% |
|  |  |  |
|  |  |  |
|  |  |  |

# Section 6 Any other section you want to add

There is no other section

# Section 7 References

There is no references

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