**SCHOOL OF DIGITAL MEDIA AND INFOCOMM TECHNOLOGY (DMIT)**

|  |  |
| --- | --- |
| **Date of Submission:** | Date of submission |

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

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

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

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 1530317 | Angelica Gonzales |
| 1530320 | Niyati Srinivasan |
| 1543007 | Eunice Poh Peck Rong |

**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](#_Toc491140230)

[A. Where we have uploaded our tutorial 3](#_Toc491140231)

[B. Why have we chosen to upload to this site 3](#_Toc491140232)

[C. What have we uploaded 3](#_Toc491140233)

[D. What is the application about? 4](#_Toc491140234)

[AnE Smart Doorbell Systems 4](#_Toc491140235)

[E. Summary of the steps that will be described 5](#_Toc491140236)

[F. How does the final RPI set-up looks like? 6](#_Toc491140237)

[G. How does the web application look like? 7](#_Toc491140238)

[Welcome 7](#_Toc491140239)

[Login 7](#_Toc491140240)

[Register 8](#_Toc491140241)

[Dashboard 8](#_Toc491140242)

[Picture 9](#_Toc491140243)

[Lights 10](#_Toc491140244)

[Message 11](#_Toc491140245)

[Section 2 Hardware Requirements 12](#_Toc491140246)

[Hardware checklist 12](#_Toc491140247)

[Button 12](#_Toc491140248)

[LED 12](#_Toc491140249)

[330Ω Resistor 13](#_Toc491140250)

[10kΩ Resistor 13](#_Toc491140251)

[I2C LCD Screen 13](#_Toc491140252)

[Buzzer 14](#_Toc491140253)

[Jumper Wires 15](#_Toc491140254)

[PiCam 15](#_Toc491140255)

[Section 3 Setup Overview 16](#_Toc491140256)

[Fritzing Diagram 16](#_Toc491140257)

[Codes 17](#_Toc491140258)

[Firebase Console 19](#_Toc491140259)

[Bluemix Console 19](#_Toc491140260)

[Section 3 AnE Smart Doorbell Systems (Part 1) 20](#_Toc491140261)

[Connect the I2C LCD Screen 20](#_Toc491140262)

[Connect the LED and Resistor 21](#_Toc491140263)

[Connect the Piezo Buzzer 22](#_Toc491140264)

[Connect the Button 23](#_Toc491140265)

[Enable Camera 24](#_Toc491140266)

[Section 4 Setting up Smart Doorbell (Part 2) 25](#_Toc491140267)

[Amazon Web Services (AWS) Setup 25](#_Toc491140268)

[Section 5 AnE Smart Doorbell Systems (Part 3) 43](#_Toc491140269)

[Installing the necessary libraries 43](#_Toc491140270)

[ane\_doorbell.py 44](#_Toc491140271)

[Section 6 AnE Smart Doorbell Systems (Part 4) 51](#_Toc491140272)

[Node.js on Bluemix 51](#_Toc491140273)

[Code 57](#_Toc491140274)

[Telegram Bot 86](#_Toc491140275)

[Firebase Console 89](#_Toc491140276)

[Publish to Bluemix 94](#_Toc491140277)

[Section 7 AnE Smart Doorbell Systems (Part 5) 95](#_Toc491140278)

[Web Application Setup 95](#_Toc491140279)

[Add AWS Certificates 96](#_Toc491140280)

[server.py 97](#_Toc491140281)

[welcome.html 102](#_Toc491140282)

[login.html 105](#_Toc491140283)

[register.html 108](#_Toc491140284)

[dashboard.html 111](#_Toc491140285)

[lcd.html 116](#_Toc491140286)

[light.html 122](#_Toc491140287)

[picture.html 129](#_Toc491140288)

[welcome.css 136](#_Toc491140289)

[login.css 141](#_Toc491140290)

[register.css 146](#_Toc491140291)

[style.css 153](#_Toc491140292)

[Section 9 155](#_Toc491140293)

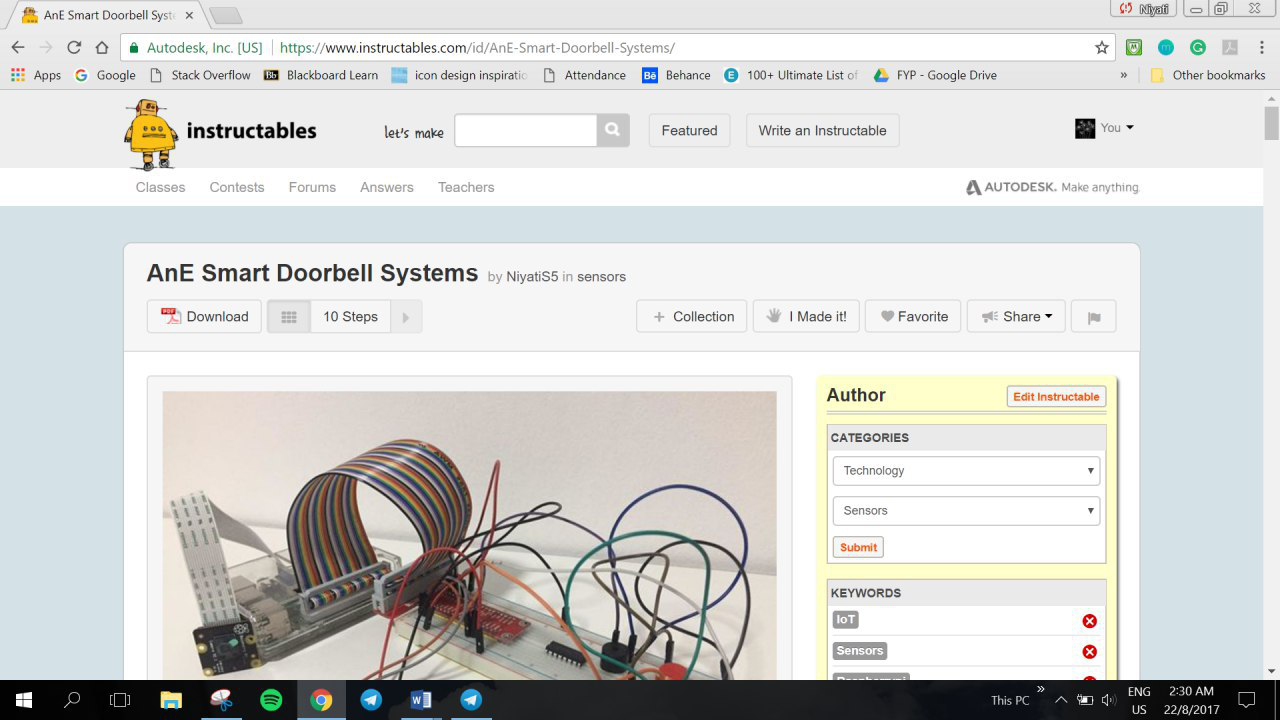
[Running 155](#_Toc491140294)

[Web Application 155](#_Toc491140295)

[Smart Doorbell 155](#_Toc491140296)

# Section 1 Overview of project

* 1. Where we have uploaded our tutorial



Link: <https://www.instructables.com/member/NiyatiS5/instructables/>

* 1. Why have we chosen to upload to this site

Instructable is an online community that allows you explore, share and make projects. It has become a known documentation system since 2005 and offers a variety of instructable classes and subjects, ranging from 3D printing to Design. It allows you to add texts, images, and files, which ensures your documents are of an expected quality before you publish your built projects, demonstrate tutorials, solicit feedback or market projects.

* 1. What have we uploaded

Instructions and images for better understanding as been uploaded to instructables.

* 1. What is the application about?

### AnE Smart Doorbell Systems

AnE Smart Doorbell Systems is a system that aims to inform you when someone presses the button of your doorbell. When a guest presses the button of the doorbell, which is placed outside the house, the buzzer will ring. The PiCam, which is mounted outside the house, allows the them to take a photo/ short video should they feel that there are potential intruder(s).



The LED is placed inside the house to allow the homeowner(s) to simulate that the house is not vacant and ward away potential intruders.

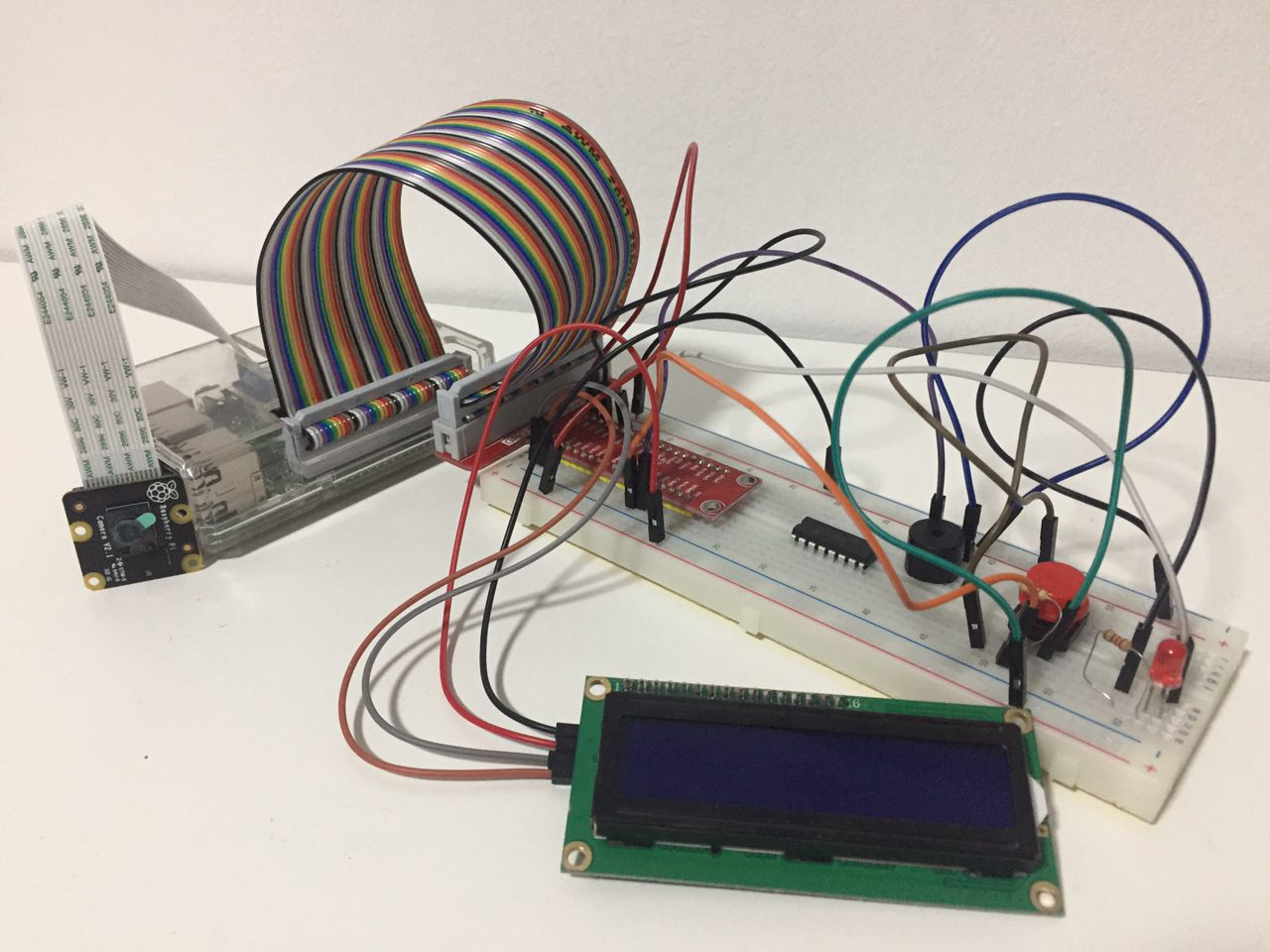
To differentiate it from your standard doorbell system, it makes use of Telegram messenger application to notify the home owner(s) when a guest presses the button and allow the home owner(s) to send a response. There are various types of responses that the homeowner(s) can send- display a message via LCD screen, turn on LED or take a video/ picture using the PiCam. Should they not have a Telegram account linked to their AnE Smart Doorbell Systems’ account, they can make use of the web interface which allows them to perform the same functions as if they were to use Telegram.

* 1. Summary of the steps that will be described

Provide a bullet list of the steps that will be covered in the other parts of this tutorial

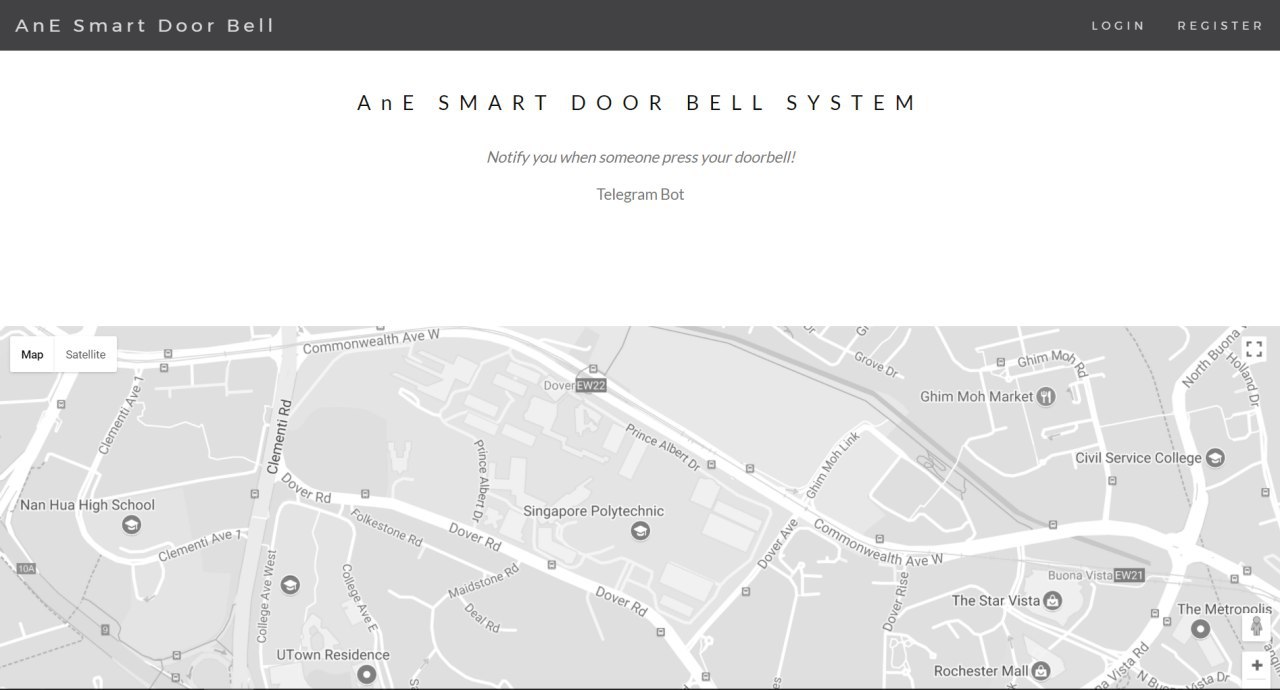
|  |  |  |
| --- | --- | --- |
|  | Section | Description |
|  | Overview | Provides an overview of what you will be doing to make this application. |
|  | Hardware Requirements | Provides an overview of hardware components required. |
|  | Setup Overview | Shows the expected outcome after setting up the application |
| Sections 4 to 8 provides the step-by-step instructions to set up the application | | |
|  | AnE Smart Doorbell Systems (Part 1) | Provides a step-by-step tutorial on setting up your hardware components. |
|  | AnE Smart Doorbell Systems (Part 2) | Provides a step-by-step tutorial on setting up your AWS IoT Console |
|  | AnE Smart Doorbell Systems (Part 3) | Provides you with the commands to install necessary libraries to set up your AnE Smart Doorbell System functions. |
|  | AnE Smart Doorbell Systems (Part 4) | Provides you with a step-by-step tutorial to setting up your Node.js Server. |
|  | AnE Smart Doorbell Systems (Part 5) | Provides a step-by-step tutorial on setting up the website |
|  | Running the System | Steps to run the Website and the Product |

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

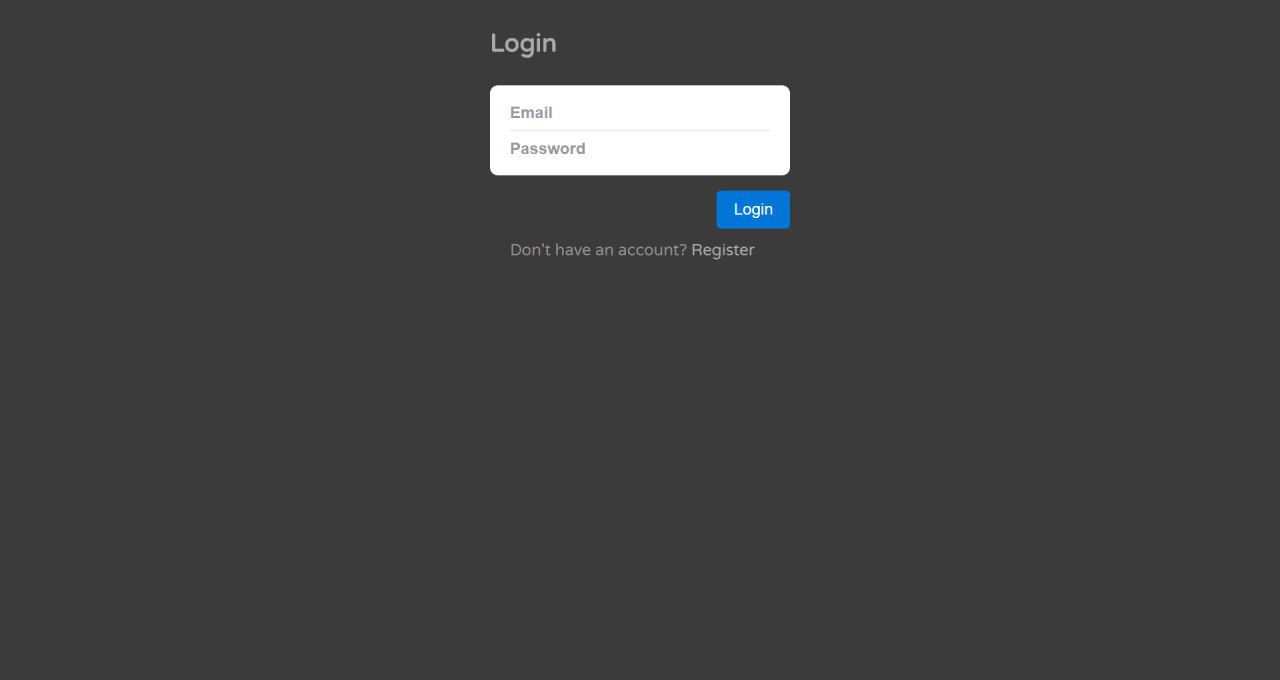


* 1. How does the web application look like?

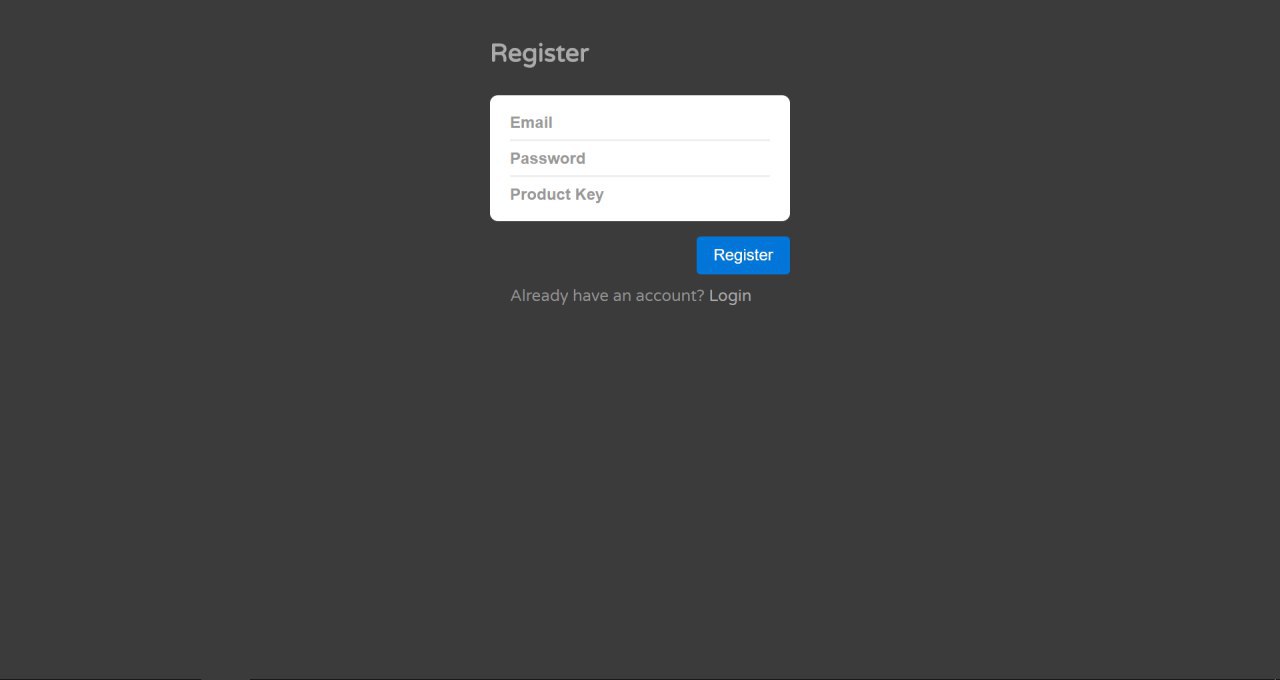
### Welcome



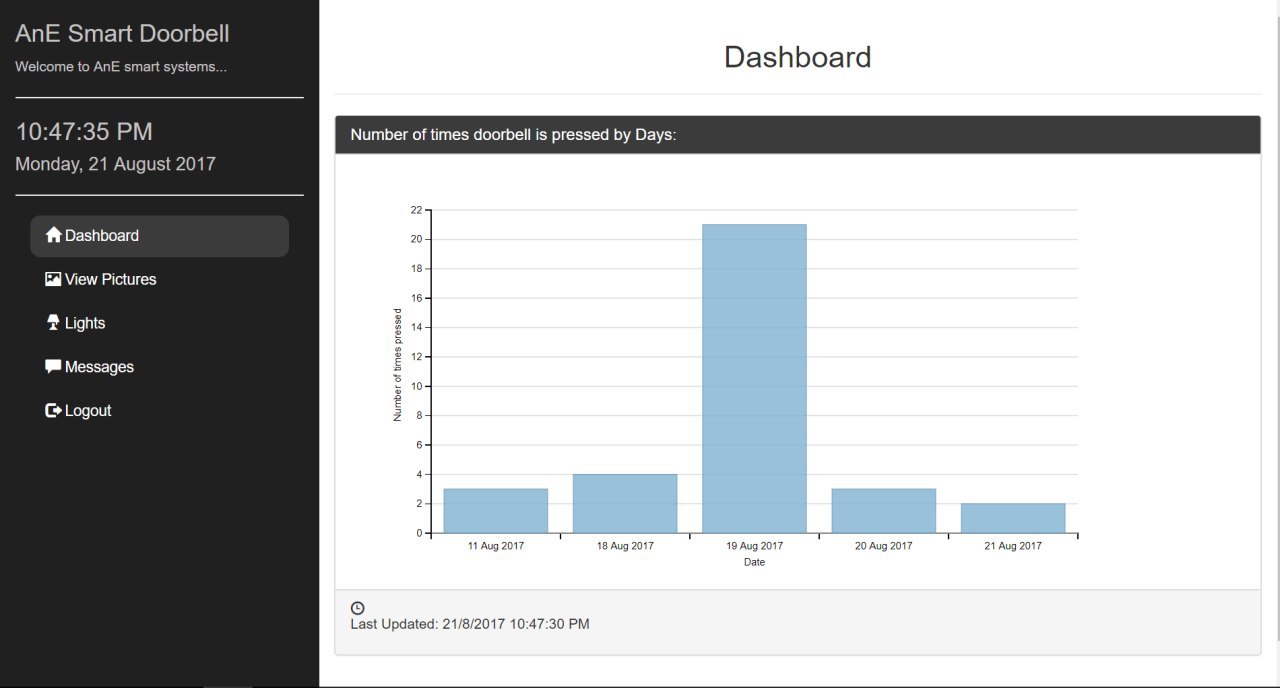
### Login



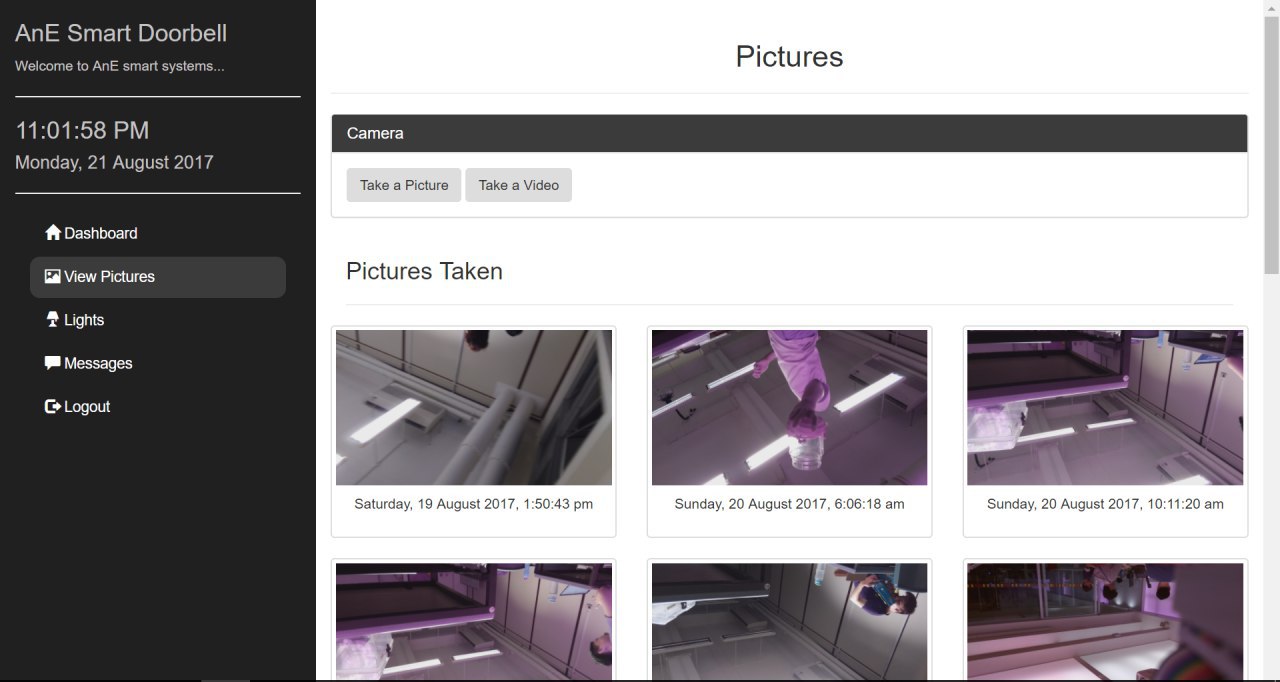
### Register

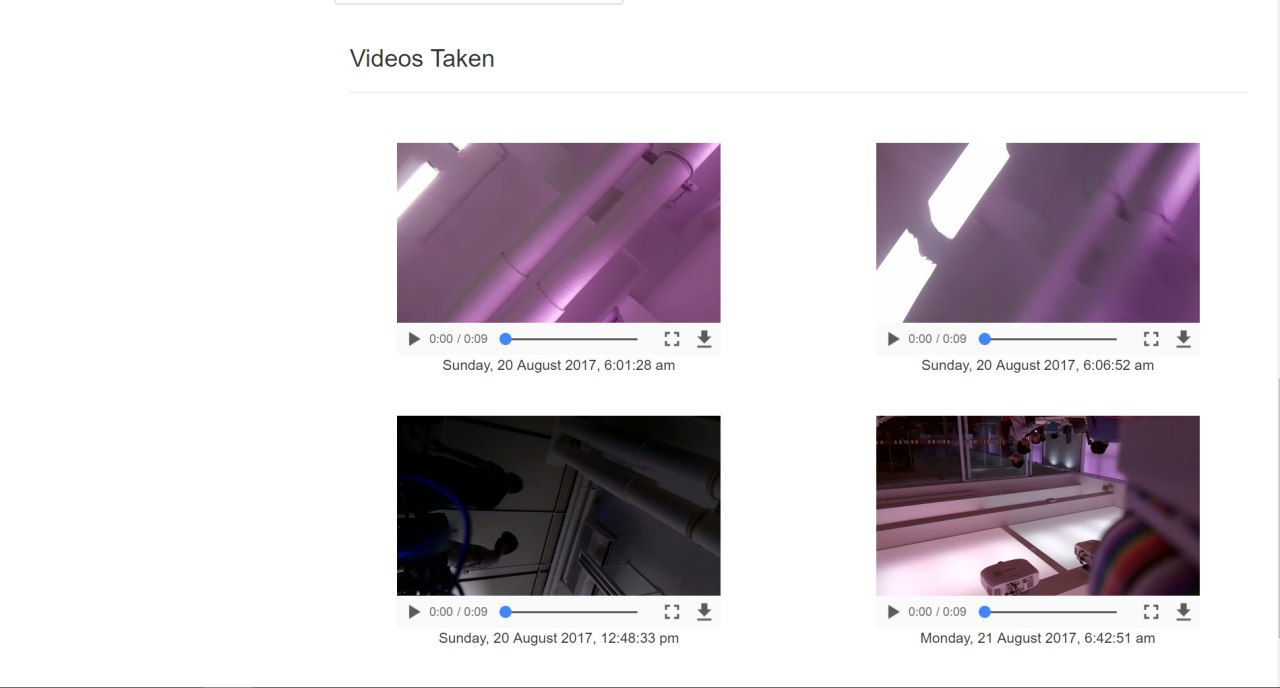


### Dashboard

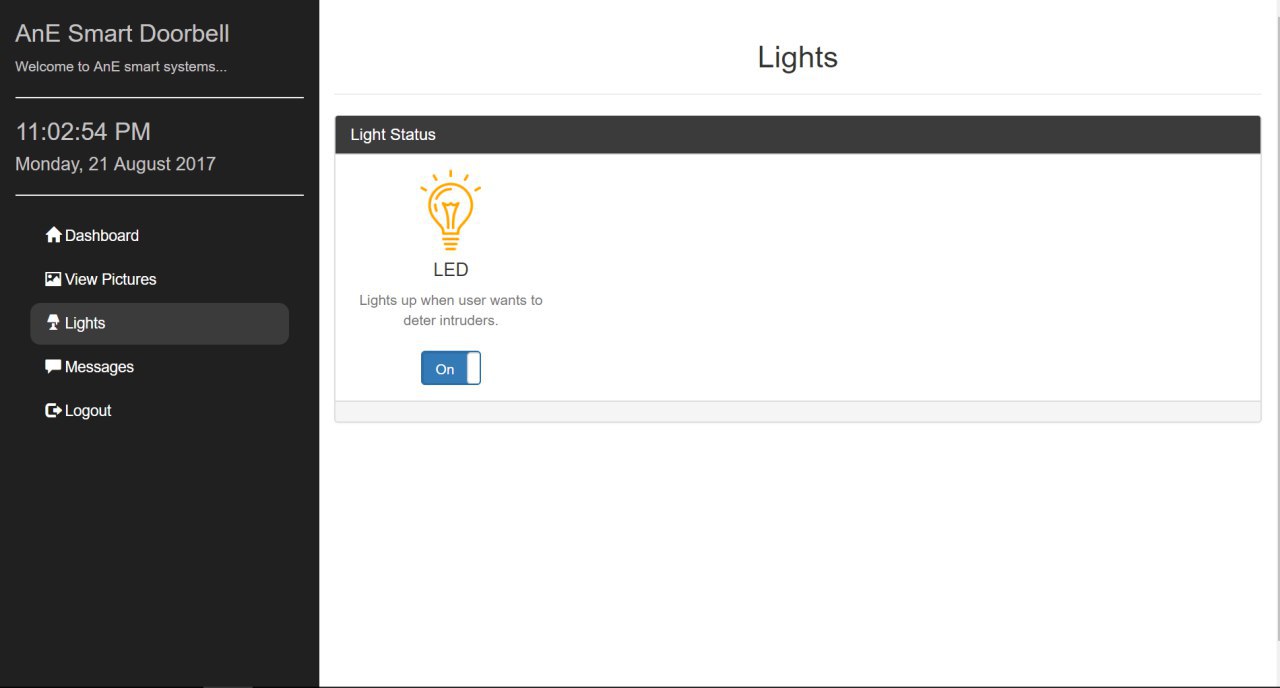


### Picture

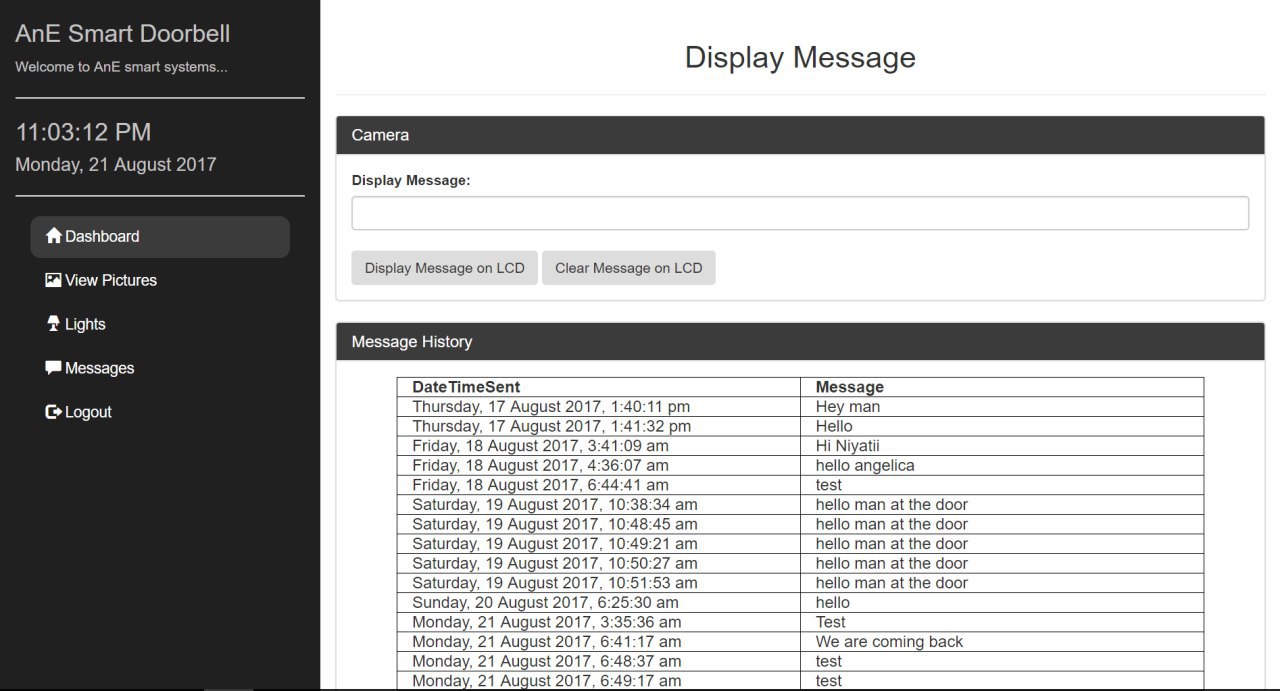




### Lights



### Message



# Section 2 Hardware Requirements

Hardware checklist

Overview of the hardware components required for this application

### Button

|  |  |
| --- | --- |
| Tactile Push-button | |
| The button contains 4 legs where signals are sent. The push buttons act as a sensor and we can determine whether it is pressed or not pressed through its High or Low state passed through the legs  In this Smart Doorbel application, the Button will represents the Doorbell where visits press on it. | Image result for tactile push-button  Button  Quantity: 1 |

### LED

|  |  |
| --- | --- |
| Light-emitting diode | |
| LEDs consists of two legs where the longer leg (anode) goes to the positive voltage and the shorter leg (cathode) goes to the negative side of the circuit  The LEDs in Smart Doorbell are placed behind the doors for users to simulate that someone is at home. | Image result for led fritzing  LED  Quantity: 1 |

### 330Ω Resistor

|  |  |
| --- | --- |
| 330ohm resistor | |
| Resistors must always be used when connecting LEDs to limit the amount of current flowing through  We will be using a 300Ω resistor which can be identified by its colour band: Orange, Orange, Brown and Gold | 330Ω Resistors  Quantity: 1 |

### 10kΩ Resistor

|  |  |
| --- | --- |
| 10k ohm Resistor | |
| To moderate the flow of current for the LDR circuit, we will be using a 10kΩ resistor which can be identified by its colour band: Brown, Black, Orange and Gold | 10kΩ Resistors  Quantity: 1 |

### I2C LCD Screen

|  |  |
| --- | --- |
| LCD Screen | |
| we will use the commonly available 16x2 LCD  which can display up to 32 characters.  We will use the i2C version which require you to connect  only 2 GPIO pins to your Raspberry Pi.  In this Smart Doorbell application, the LCD screen will show the users message to the guests at the door. | Image result for I2C LCD Screen  LCD Screen  Quantity: 1 |

### Buzzer

|  |  |
| --- | --- |
| Buzzer | |
| A buzzer is an audio signaling device which is commonly found in circuits to create a buzzing or beeping noise.  Buzzers can be categorized as active buzzers and passive ones.  An active buzzer can be connected just like an LED but they are even easier to use because a resistor is not needed.  A buzzer typically has 2 pins o VOUT – Connect this to a GPIO pin to control its value o GND – Connect this to ground | 5V Active Buzzer Packs of 1, 2 or 5 Continuous Tone  Buzzer  Quantity: 1 |

### Jumper Wires

|  |  |
| --- | --- |
| Jumper Wires | |
| Jumper wires are electrical wires which are needed to connect the components together. | Image result for jump wires types  M/F Jumper Wires  Quantity: 4  M/M Jumper Wires  Quantity: 8 |

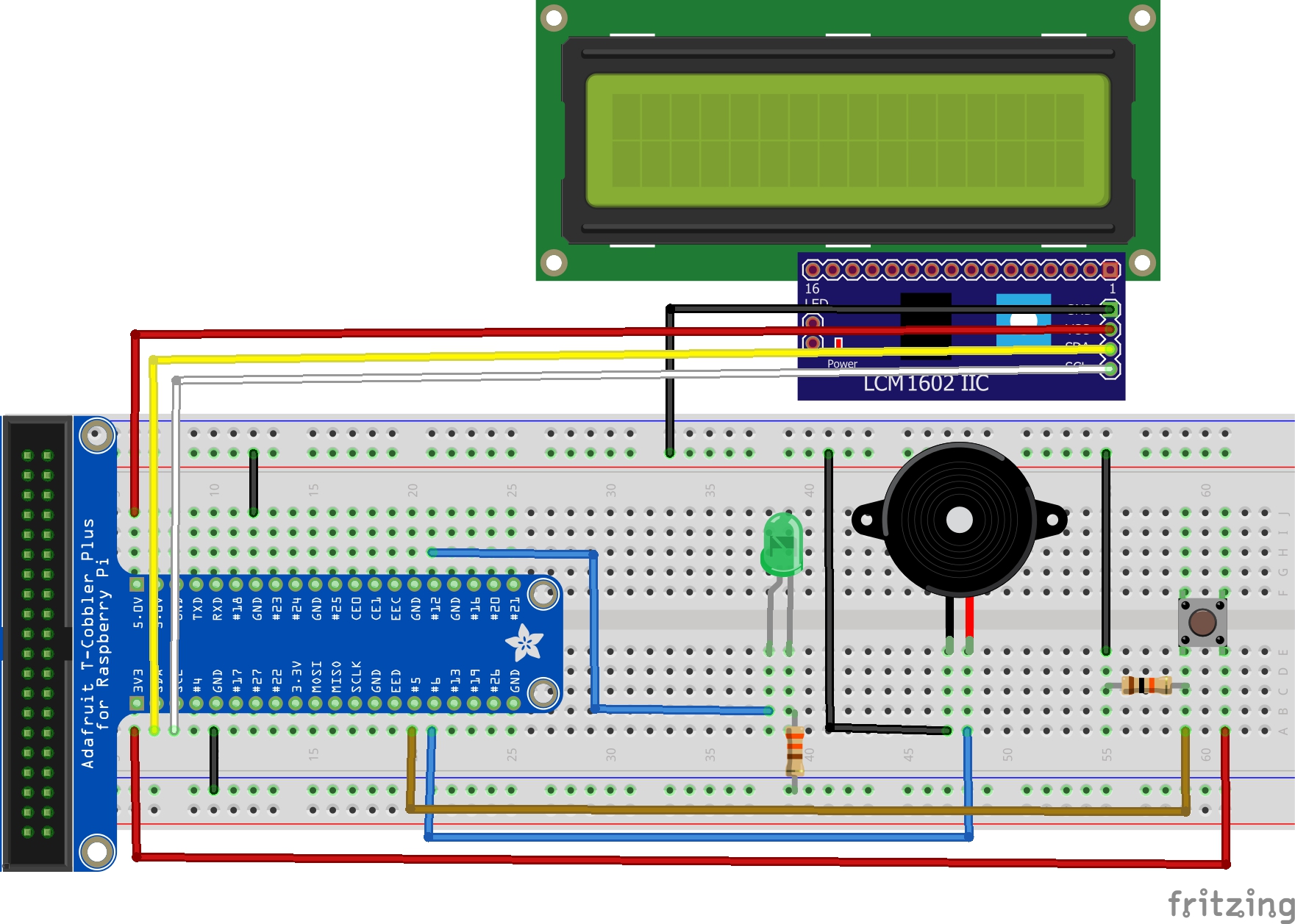
### PiCam

|  |  |
| --- | --- |
| PiCam | |
| PiCam allows the homeowner(s) to take a photo or a video to monitor for potential intruders | PiCam  Quantity: 1 |

# Section 3 Setup Overview

Fritzing Diagram

This is how your final set-up would look like.



Codes

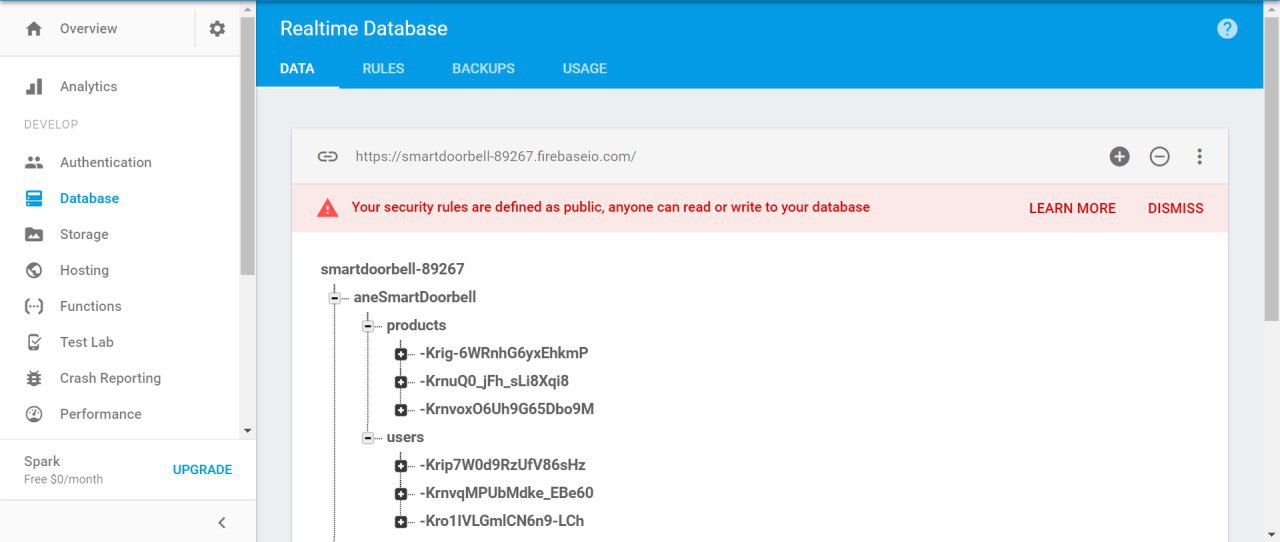
At the end of the setup, you should have the following folders created for each component, and the directory tree should look like this

|  |
| --- |
| Directory for Raspberry Pi Codes |
| .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  ├── pictures  ├── videos  ├── ane\_doorbell.py  └── awscredentials.py |

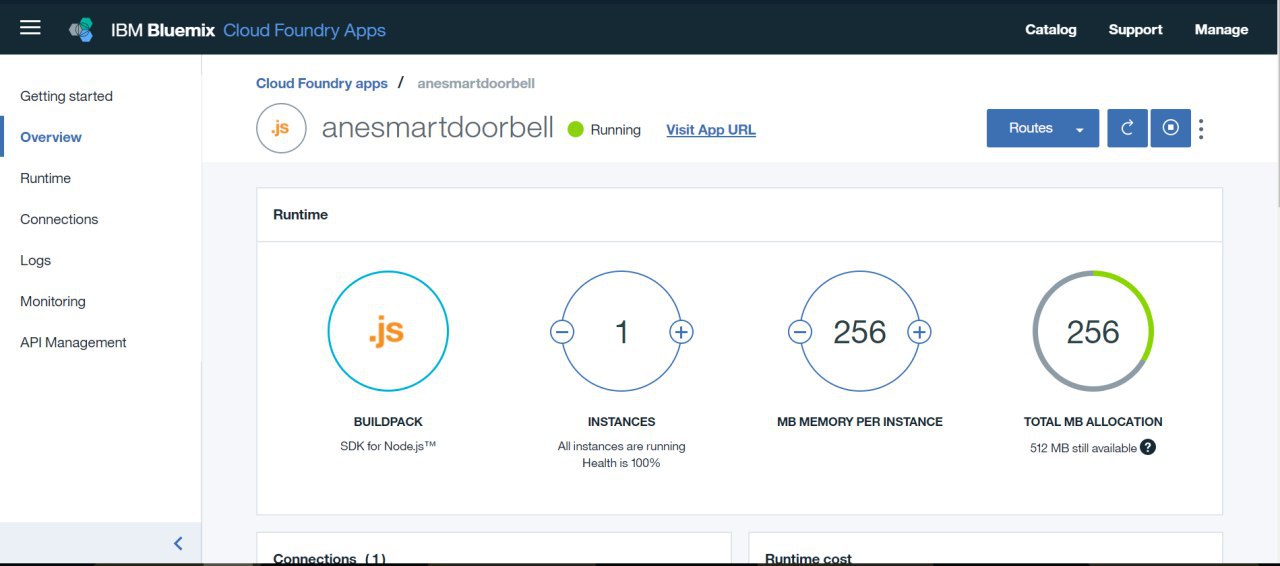
|  |
| --- |
| Directory for Node.js Server Codes |
| .  |  ├── .git  |  ├── APIs  | ├── email.js  | ├── product.js  | ├── pubsub.js  | └── user.js  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── Controllers  | ├── DeviceAPIController.js  | ├── ProductAPIController.js  | └── UserAPIController.js  |  ├── node\_modules  |  ├── .cfignore  ├── .gitignore  ├── .project  ├── app.js  ├── config.js  ├── LICENSE  ├── manifest.yml  ├── package.json  └── README.md |

|  |
| --- |
| Directory for Website Code |
| .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── static  | └── css  | ├── login.css  | ├── register.css  | ├── styles.css  | └── welcome.css  |  ├── templates  | ├── dashboard.html  | ├── lcd.html  | ├── light.html  | ├── login.html  | ├── picture.html  | ├── register.html  | └── welcome.html  |  └── server.py |

Firebase Console

****

Bluemix Console

****

# Section 3 AnE Smart Doorbell Systems (Part 1)

Connect the I2C LCD Screen

|  |
| --- |
| Connect LCD Screen |
| * + 1. Connect the pins on the LCD to the RPI as follows |
| * + 1. Install the rpi-lcd library using the commands below   sudo pip install rpi-lcd |
| * + 1. Final Setup |

Connect the LED and Resistor

|  |
| --- |
| Connect LED and Resistor |
| 1. Plug in the longer end of the LED closer to the T-Cobbler Kit |
| 1. Connect the longer end of the led using an blue jump wire to the RPi pin as follows: |
| 1. Connect the one end of the **330 Ω resistor** to the negative bus strip. 2. Connect the other end of the 330 Ω resistor in the same column as the shorter end of the LED 3. Connect one end of the black jumper wire to the GND and the other to the negative bus strip that the resistor is connected |
|  |

Connect the Piezo Buzzer

|  |
| --- |
| Steps |
| 1. Connect the Buzzer with the RPi as follows: |
|  |

Connect the Button

|  |
| --- |
| Steps |
| 1. Place the button on the breadboard as shown: |
|  |

Enable Camera

|  |
| --- |
| Enable Camera- using Graphical User Interface (GUI) |
| 1. Open the Raspberry Pi Configuration Tool from the main menu |
| 2. Select Interfaces, enable Camera and click OK |

# Section 4 Setting up Smart Doorbell (Part 2)

Amazon Web Services (AWS) Setup



AnE Smart Doorbell Systems makes use of Amazon Web Services (AWS) to publish and subsribe to the state of the hardware components, such as LCD Screen and PiCam.

­­­When the button is pressed, it publishes a payload, which informs subscribers such Telegram or web app. The other hardware components also publishes payloads, which keeps the user informed of its state and keeps track of its history.

|  |
| --- |
| Sign In to the AWS IoT Console |
| 1. At the AWS Console, click on Services |
| 1. Select AWS IoT     You will be navigated here |

|  |
| --- |
| Create & Register Thing |
| 1. At the AWS IoT Dashboard, select Registry | Things |
| 1. Click on Create   \*Note: If you don’t have a thing yet, click on Register a thing. |
| 1. Give you Thing a name and click Create thing |
| 1. Select Interact and copy and paste REST API Endpoint to a notepad |

|  |
| --- |
| Create Certificates |
| 1. Click the back arrow to go back to AWS IoT Dashboard |
| 1. At the AWS IoT Dashboard, select Security | Certificates |
| 1. Click on Create   \*Note: If you don’t have a certificate yet, click on Create certificate |
| 1. Click Download for each certificate |
| 1. When you click Download for your root CA, right click and save as rootca.pem |
| 1. Move your certs to a new folder named certs and rename them accordingly |
| 1. Click Activate |

|  |
| --- |
| Create Security Policy |
| 1. Click Attach a policy |
|  |
| 2. Click Create a new policy |
| 3. Define the policy name and authorized actions |
| 4. Click Create |

|  |
| --- |
| Attach Security Policy and Thing to Certificates |
| 1. Click the back arrow to go back to AWS IoT Dashboard |
| 2. At the AWS IoT Dashboard, select Security | Certificates |
| 3. Click on the Certificate menu and select Attach policy |
| 4. Select the policy you created and click Attach |
| 5. Click on the Certificate menu and select Attach thing |
| 6. Select the thing you created and click Attach |

|  |
| --- |
| Create A Folder |
| 1. Open a terminal in your raspberryPi and run   mkdir ~/smartdoorbell |
| 2. Transfer certs folder to your raspberryPi and put it inside ~/smartdoorbell folder |

|  |
| --- |
| smartdoorbell |
| .  |  └── certs  ├── certificate.pem.crt  ├── private.pem.key  ├── public.pem.key  └── rootca.pem |

|  |
| --- |
| awscredentials.py |
| 1. Create **smartdoorbell/awscredentials.py** to store your AWS Credentials |
| 1. Copy and Paste the following codes  |  | | --- | | accessKey = '' #specify your AWS access key  secretAccessKey = '’ #specify your AWS secret access key | |
| 1. Your directory should look like this now |
| .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  └── awscredentials.py |

# Section 5 AnE Smart Doorbell Systems (Part 3)

Installing the necessary libraries

|  |
| --- |
| Steps |
| 1. Install Flask with this command  sudo pip install flask |
| 2. Install the AWS Python library with this command  sudo pip install AWSIoTPythonSDK |
| 3. Type the following command to install Boto, the Python library for AWS on your raspberrypi  sudo pip install boto3 |
| 4. Type the following command to install the AWS Command‐Line Interface Client on your Raspberry Pi  sudo pip install awscli |
| 5. Create a python script ane\_doorbell.py with the code below   |  | | --- | | sudo nano ~/smartdoorbell/ane\_doorbell.py | |

|  |
| --- |
| Create folders |
| 1. Open a terminal in your raspberryPi and run   mkdir ~/smartdoorbell/pictures |
| 1. Open a terminal in your raspberryPi and run   mkdir ~/smartdoorbell/videos |

ane\_doorbell.py

|  |
| --- |
| Steps |
| 1. Create a python script **ane\_doorbell.py** with the code below  |  | | --- | | sudo nano ~/smartdoorbell/ane\_doorbell.py | |
| 1. Copy the following codes  |  | | --- | | # coding: utf8  from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient  from gpiozero import Button, Buzzer, LED  from rpi\_lcd import LCD  from picamera import PiCamera  from signal import pause  from datetime import datetime  import calendar  from time import sleep  import json  import boto3  import botocore  import time  import subprocess  import shlex  import os  import awscredentials  camera = PiCamera()  bz = Buzzer(6)  led = LED(18)  lcd = LCD()  button = Button(5, pull\_up=False)  # Set the filename and bucket name  bucket\_name = 'anesmartdoorbell' # replace with your own unique bucket name  region = 'ap-southeast-1'  exists = True  spam = False  start = time.time()  host = "a3t0nm0d0jd547.iot.us-west-2.amazonaws.com"  rootCAPath = "certs/rootca.pem"  certificatePath = "certs/certificate.pem.crt"  privateKeyPath = "certs/private.pem.key"  accessKey = awscredentials.accessKey  secretAccessKey = awscredentials.secretAccessKey  productKey = "A1B2C3"  # Create an S3 resource  s3 = boto3.resource('s3', aws\_access\_key\_id=accessKey,  aws\_secret\_access\_key=secretAccessKey)  def onLight():  led.on()  def offLight():  led.off()  def lcdWrite(message):  lcd.text(message[:16], 1)  lcd.text(message[16:32], 2)  def lcdClear():  lcd.clear()  def doorbellPressed():  global productKey, start, spam  bz.on()  if (spam == False):  takePicture()  start = time.time()  d = datetime.utcnow()  timestamp = calendar.timegm(d.utctimetuple())  payload = json.dumps({  "device": "button",  "productKey": productKey,  "action": "button\_pressed",  "data": {  "datetime": timestamp  }  })  my\_rpi.publish("anedingding/" + productKey + "/button", payload, 1)  spam = True  else:  difference = time.time() - start  if(difference > 10):  spam = False  start = time.time()  def doorbellRelease():  bz.off()  def takePicture():  global region, productKey, bucket\_name  print("taking picture")  d = datetime.utcnow()  timestamp = calendar.timegm(d.utctimetuple())  file\_name = "{}.jpg".format(timestamp)  camera.capture("pictures/" + file\_name)  full\_path = "{}/{}".format(productKey, file\_name)  # Upload picture to s3  s3.Bucket(bucket\_name).put\_object(Key=full\_path,  Body=open("pictures/" + file\_name, 'rb'))  picturePath = "".join(  ["https://s3-", region, ".amazonaws.com/", bucket\_name, "/", full\_path])  payload = json.dumps({  "device": "camera",  "productKey": productKey,  "action": "picture\_taken",  "data": {  "datetime": timestamp,  "picturePath": picturePath  }  })  my\_rpi.publish("anedingding/" + productKey + "/camera", payload, 1)  print("File uploaded")  def takeVideo():  global region, productKey, bucket\_name  d = datetime.utcnow()  timestamp = calendar.timegm(d.utctimetuple())  file\_name = "{}.h264".format(timestamp)  new\_file\_name = "{}.mp4".format(timestamp)  full\_path = "{}/{}".format(productKey, new\_file\_name)  camera.start\_recording("videos/" + file\_name)  sleep(8)  camera.stop\_recording()  video\_save\_path = os.path.dirname(os.path.realpath(  \_\_file\_\_)) + "/videos/{}".format(file\_name)  # Convert video to mp4 format  from subprocess import CalledProcessError  command = "MP4Box -add {} videos/{}".format(video\_save\_path, new\_file\_name)  try:  output = subprocess.check\_output(  command, stderr=subprocess.STDOUT, shell=True)  subprocess.Popen("rm {}".format(video\_save\_path),  shell=True) # Delete the h264 video  except CalledProcessError as e:  print('FAIL:\ncmd:{}\noutput:{}'.format(e.cmd, e.output))  # Upload video to s3  s3.Bucket(bucket\_name).put\_object(Key=full\_path,  Body=open("videos/" + new\_file\_name, 'rb'))  videoPath = "".join(  ["https://s3-", region, ".amazonaws.com/", bucket\_name, "/", full\_path])  payload = json.dumps({  "device": "camera",  "productKey": productKey,  "action": "video\_taken",  "data": {  "datetime": timestamp,  "videoPath": videoPath  }  })  my\_rpi.publish("anedingding/" + productKey + "/camera", payload, 1)  def customCallback(client, userdata, message):  global productKey  print("Received a new message: ")  print(json.loads(message.payload))  print("from topic: {}".format(message.topic))  print("‐‐‐‐‐‐‐‐‐‐‐‐‐‐\n\n")  payloadData = json.loads(message.payload)  action = payloadData["action"]  device = payloadData["device"]  if (device == "led"):  if(action == "on\_led"):  onLight()  payload = json.dumps({  "device": "led",  "productKey": productKey,  "action": "led\_on"  })  my\_rpi.publish("anedingding/" + productKey + "/led", payload, 1)  elif (action == "off\_led"):  offLight()  payload = json.dumps({  "device": "led",  "productKey": productKey,  "action": "led\_off"  })  my\_rpi.publish("anedingding/" + productKey + "/led", payload, 1)  elif (device == "lcd"):  if (action == "display\_message"):  message = str(payloadData["data"]["message"])  lcdWrite(message)  payload = json.dumps({  "device": "lcd",  "productKey": productKey,  "action": "message\_display",  "data": {  "message": message,  "datetime": str(payloadData["data"]["datetime"])  }  })  my\_rpi.publish("anedingding/" + productKey + "/lcd", payload, 1)  elif (action == "clear\_message"):  lcdClear()  payload = json.dumps({  "device": "lcd",  "productKey": productKey,  "action": "message\_clear"  })  my\_rpi.publish("anedingding/" + productKey + "/lcd", payload, 1)  elif (device == "camera"):  if (action == "take\_picture"):  takePicture()  elif (action == "take\_video"):  takeVideo()  try:  s3.meta.client.head\_bucket(Bucket=bucket\_name)  except botocore.exceptions.ClientError as e:  error\_code = int(e.response['Error']['Code'])  if error\_code == 404:  exists = False  if exists == False:  # Create the bucket policy  bucket\_policy = json.dumps({  "Version": "2008-10-17",  "Statement": [  {  "Sid": "AllowPublicRead",  "Effect": "Allow",  "Principal": {  "AWS": "\*"  },  "Action": "s3:GetObject",  "Resource": "arn:aws:s3:::%s/\*/\*" % bucket\_name  }  ]  })  s3.create\_bucket(Bucket=bucket\_name, CreateBucketConfiguration={  'LocationConstraint': region})  # Create an S3 client  s3client = boto3.client('s3', aws\_access\_key\_id=accessKey,  aws\_secret\_access\_key=secretAccessKey)  # Set the new policy on the given bucket  s3.put\_bucket\_policy(Bucket=bucket\_name, Policy=bucket\_policy)  print("S3 bucket created")  my\_rpi = AWSIoTMQTTClient(productKey)  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("anedongdong/" + productKey + "/#", 1, customCallback)  print("Subscribed")  button.when\_pressed = doorbellPressed  button.when\_released = doorbellRelease  pause() | |

|  |
| --- |
| Task |
| 1. Create 2 folders **pictures** and **videos** in your project directory for storing the pictures and videos taken using the PI Camera |
| 1. You project directory should look like this now   .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  ├── pictures  ├── videos  ├── ane\_doorbell.py  └── awscredentials.py |

# Section 6 AnE Smart Doorbell Systems (Part 4)

Node.js on Bluemix

|  |
| --- |
| Overview |
| * + 1. Create a Bluemix Cloud Foundry App     2. Enable Continuous Integration     3. Clone the files in Git to your computer |

|  |
| --- |
| Task |
| 1. In you Bluemix Dashboard, click on Create App |
| 1. Search “SDK for Node.js” and click on SDK for Node.js under the Cloud Foundry Apps |
| 1. Key in the App name for your App and click Create |
| 1. After the App has been created, scroll down to Continuous delivery under Overview tab and click Enable |
| 1. Ensure that the settings are the same and click Create |
| 1. At the Overview of your Toolchains, click on Git |
| 1. You will be navigated to your project in Git. Click on the Blue pattern circle on the top right hand corner and click on Settings |
| 1. In the Access Token tab create a Personal Access Token for yourself to clone the project |
| 1. Copy the generated Access Token in a notepad, we wil be using it when cloning the project later |
| 1. Copy your **Git URL** and take note of the **username** below |
| 1. Open your command prompt in the directory you want to save the project in and type in   git clone <project url>  Press enter and you will be prompted to key in the Git Credentials.  Enter the **UserName** from the Git Console and the generated **Personal Access Token** as the Password |
| 1. Done the Node.js project is now in your project directory |

### Code

|  |
| --- |
| The Project you clone should look like this |
| .  |  ├── .git  ├── node\_modules  ├── public  | ├── images  | | └── newapp-icon.png  | ├── stylesheets  | | └── style.css  | └── index.html  ├── .cfignore  ├── .gitignore  ├── .project  ├── app.js  ├── config.js  ├── LICENSE  ├── manifest.yml  ├── package.json  └── README.md |

|  |
| --- |
| Task |
| 1. Open the project in a text editor such as Visual Studio Code or Sublime |
| 1. Replace the **package.json** file with the codes below  |  | | --- | | {  "name": "test",  "version": "0.0.0",  "private": true,  "scripts": {  "start": "node app"  },  "dependencies": {  "aws-iot-device-sdk": "^2.0.1",  "body-parser": "~1.16.0",  "cors": "^2.8.4",  "express": "~4.14.1",  "firebase": "^4.2.0",  "moment": "^2.18.1",  "node-telegram-bot-api": "^0.28.0",  "nodemailer": "^4.0.1",  "ws": "^3.1.0"  }  } | |
| 1. Run npm install –save to install the packages needed for this project |
| 1. Replace the app.js file with the codes below  |  | | --- | | var express = require('express');  var path = require('path');  var bodyParser = require('body-parser');  var cors = require('cors');  var firebase = require('firebase');  var app = express();  // For BodyParser  app.use(bodyParser.json());  app.use(bodyParser.urlencoded({ extended: false }));  app.use(cors());  var port = process.env.VCAP\_APP\_PORT || 3000;  // Initialise Firebase  var firebaseCredentials = require(path.resolve('./config.js')).firebaseCredentials;  firebase.initializeApp(firebaseCredentials);  // Routes  var user = require('./Controllers/UserAPIController');  var product = require('./Controllers/ProductAPIController');  var device = require('./Controllers/DeviceAPIController');  app.use('/api/user', user);  app.use('/api/product', product);  app.use('/api/device', device);  // AWS MQTT and Telegram listener  require(path.resolve('./APIs/pubsub'));  // Catch 404 and forward to error handler  app.use(function (req, res, next) {  var err = new Error('Not Found');  err.status = 404;  next(err);  });  app.listen(port);  console.log('Magic happens on port ' + port);  module.exports = app; | |
| 1. Create a folder call **certs** for storing the AWS certificates from the AWS IOT Console. Your directory for the certs should look like this  |  | | --- | | .  |  └── certs  ├── certificate.pem.crt  ├── private.pem.key  ├── public.pem.key  └── rootca.pem | |

|  |
| --- |
| Task |
| 1. Create a Controllers folder where all the API routes will be.  |  | | --- | | .  |  └── Controllers | |
| 1. Create a javascript file **DeviceAPIController.js** and paste the code inside the file  |  | | --- | | var express = require('express');  var router = express.Router();  var path = require('path');  var user = require(path.resolve('./APIs/user.js'));  /\*\*  \* [GET]  \* Return list of photos taken  \* http://localhost:8680/api/device/photo  \*/  router.get('/photo', function (req, res) {  user.getUserPhoto(req.headers.userkey).then(function (result) {  res.send({ success: true, data: result })  })  })  /\*\*  \* [GET]  \* Return list of videos taken  \* http://localhost:8680/api/device/video  \*/  router.get('/video', function (req, res) {  user.getUserVideo(req.headers.userkey).then(function (result) {  res.send({ success: true, data: result })  })  })  /\*\*  \* [GET]  \* Return list of button pressed info  \* http://localhost:8680/api/device/button  \*/  router.get('/button', function (req, res) {  user.getUserButton(req.headers.userkey).then(function (result) {  res.send({ success: true, data: result })  })  })  /\*\*  \* [GET]  \* Return list of displayed messages  \* http://localhost:8680/api/device/message  \*/  router.get('/message', function (req, res) {  user.getUserMessages(req.headers.userkey).then(function (result) {  res.send({ success: true, data: result })  })  })  /\*\*  \* [GET]  \* Return status of led  \* http://localhost:8680/api/device/led  \*/  router.get('/led', function (req, res) {  data.getUserLight(req.headers.userkey).then(function (result) {  res.send({ success: true, data: result })  })  })  module.exports = router | |
| 1. Create a javascript file **ProductAPIController.js** and paste the code inside the file  |  | | --- | | var express = require('express');  var router = express.Router();  var path = require('path');  var product = require(path.resolve('./APIs/product.js'));  /\*\*  \* [POST]  \* Add Product Key to firebase  \* http://localhost:3000/api/product/  \* Body: JSON(application/json)  \* {  "productKey": ""  }  \*/  router.post('/', function (req, res) {  var productKey = req.body.productKey;  product.addProductKey(productKey)  .then(function (result) {  res.send({ success: true, message: result });  }).catch(function (error) {  res.send({ success: false, message: error });  });  });  module.exports = router; | |
| 1. Create a javascript file **UserAPIController.js** and paste the code inside the file  |  | | --- | | var express = require('express');  var router = express.Router();  var path = require('path');  var user = require(path.resolve('./APIs/user.js'));  var config = require('../config').others;  router.get('/', function (req, res) {  res.send('Available')  })  /\*\*  \* [POST]  \* Create Account for Users using Firebase Authentication  \* http://localhost:3000/api/user/signup  \* Body: JSON(application/json)  \* {  "email": "",  "password": "",  "productKey": ""  }  \*/  router.post('/signup', function (req, res) {  var email = req.body.email;  var password = req.body.password;  var productKey = req.body.productKey;  console.log(productKey)  user.signup(email, password, productKey)  .then(function (result) {  res.send({ success: true, message: result });  }).catch(function (error) {  res.send({ success: false, message: error });  });  });  /\*\*  \* [POST]  \* Sign in using Firebase Authentication  \* http://localhost:3000/api/user/signin  \* Body: JSON(application/json)  \* {  "email": "",  "password": ""  }  \*/  router.post('/signin', function (req, res) {  var email = req.body.email;  var password = req.body.password;  user.signin(email, password)  .then(function (result) {  res.send({ success: true, message: result.message, userKey: result.userid, productKey: result.productkey });  }).catch(function (error) {  res.send({ success: false, message: error });  });  });  /\*\*  \* [GET]  \* URL that user receive in their email to verify their telegram account  \* http://localhost:8680/verifyemail?productId=&chatId=  \*/  router.get('/verifyuser?', function (req, res) {  var productId = req.query.productId;  var chatId = req.query.chatId;  user.updateTelegramId(productId, chatId)  .then(function (result) {  res.send(result);  })  });  module.exports = router; | |

|  |
| --- |
| Task |
| 1. Create a APIs folder where all the methods will be.  |  | | --- | | .  |  └── APIs | |
| 1. Create a javascript file **email.js** and paste the code inside the file   This file contains the code to send email to user via gmail   |  | | --- | | var email = module.exports = {};  var path = require("path");  var nodemailer = require("nodemailer");  var gmailConfig = require(path.resolve('./config')).gmail;  // create reusable transporter object using the default SMTP transport  let transporter = nodemailer.createTransport({  service: 'Gmail',  auth: {  user: gmailConfig.email,  pass: gmailConfig.password  }  });  // create reusable transporter object using the default SMTP transport  email.sendEmail = function (email, productId, chatId) {  return new Promise(function (resolve, reject) {  var host = "https://anesmartdoorbell.au-syd.mybluemix.net"  link = host + "/api/user/verifyuser?productId=" + productId + "&chatId=" + chatId;  let mailOptions = {  to: email,  subject: 'Link Telegram Account', // Subject line  html: 'Hi ' + email + ',<br><br>' +  '<p>Please Click on the link below to link your Product to your Telegram account.<br>' +  '<a href=' + link + '>Link telegram account</a></p><br><br>' +  'Best Regards,<br><br>' +  'AnE Smart Doorbell Team'  };  // send mail with defined transport object  transporter.sendMail(mailOptions, (error, info) => {  if (error) {  console.log(error);  reject(error.toString())  }  resolve("Successfully sent email");  });  });  } | |
| 1. Create a javascript file **product.js** and paste the code inside the file   They file contains the methods related to Fireabse products node   |  | | --- | | var product = module.exports = {};  var path = require('path');  var firebase = require('firebase');  var rootRef = firebase.database().ref('aneSmartDoorbell');  var productRef = rootRef.child("products");  // add new product key to firebase db  product.addProductKey = function (key) {  return new Promise(function (resolve, reject) {  var newProduct = {  "productKey": key,  "userKey": ""  };  productRef.push(newProduct, function (error) {  if (error) {  console.log("Error: " + error);  reject(error);  } else {  resolve("Product Key added");  }  });  });  }  // link the product key to a user  product.linkProductToUser = function (productKey, userKey) {  productRef.once('value', function (snapshot) {  snapshot.forEach(function (oneProduct) {  if (oneProduct.val().productKey == productKey) {  productRef.child(oneProduct.key).child("userKey").set(userKey);  }  });  });  }  // Check whether product key is already linked to a user  product.checkProductKey = function (productKey) {  return new Promise(function (resolve, reject) {  productRef.once('value', function (snapshot) {  var available = "Product Key invalid";  snapshot.forEach(function (oneProduct) {  if (oneProduct.val().productKey == productKey) {  available = "That Product is already registered under another account";  if (oneProduct.val().userKey == "") {  available = true;  }  }  });  resolve(available);  });  });  } | |
| 1. Create a javascript file **pubsub.js** and paste the code inside the file   This file contains the code that publish and subscribe to AWS IoT and Telegram Bot   |  | | --- | | var pubsub = module.exports = {};  var awsIot = require('aws-iot-device-sdk');  var TelegramBot = require('node-telegram-bot-api');  var path = require("path");  var firebase = require('firebase');  var config = require(path.resolve('./config')).aws;  var telegramToken = require(path.resolve('./config.js')).telegram.token;  var email = require(path.resolve('./APIs/email.js'));  var user = require(path.resolve("./APIs/user.js"));  var product = require(path.resolve("./APIs/product.js"));  var telegram = new TelegramBot(telegramToken, { polling: true });  var rootRef = firebase.database().ref('aneSmartDoorbell');  var userRef = rootRef.child('users');  var productRef = rootRef.child('products');  // send telegram message  pubsub.sendMessage = function (chatId, message, options) {  telegram.sendMessage(chatId, message, options)  .catch(function (error) {  console.log(error)  });  }  //Listen for any kind of message.There are different kinds of  //messages.  telegram.on('message', (msg) => {  var chatId = msg.chat.id;  var userMsg = msg.text;  if (userMsg.match(/\/link (.+)|\/link/)) { // check product key availble and link it to user by sending an email to them  var productKey = userMsg.match(/\/link (.+)|\/link/)[1];  if (productKey) {  product.checkProductKey(productKey)  .then(function (available) {  if (available == true) { // no user link to product  pubsub.sendMessage(chatId, 'An account have not been created for that ProductKey. Please create an account before linking your telegram account to the product.', {});  } else if (available == "Product Key invalid") { //no such product keys  pubsub.sendMessage(chatId, available, {});  } else { // product key is linked to a user  userRef.on('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  if (oneUser.val().telegramChatId == "") {  //send email  email.sendEmail(oneUser.val().email, productKey, chatId).then(function () {  pubsub.sendMessage(chatId, 'A verification email have been sent to the email linked to that ProductKey', {});  })  } else {  pubsub.sendMessage(chatId, 'That Product is already linked to another Telegram Account', {});  }  }  })  })  }  })  } else {  pubsub.sendMessage(chatId, 'Try /link <Product Key>', {});  }  } else {  user.checkTelegramId(chatId).then(function (isLink) { // check if user is already a user  if (isLink) {  if (userMsg.match(/\/start/)) {  pubsub.sendMessage(chatId, 'Hi, I am AnE Smart Doorbell Bot.\n\n'  + 'Link your Smart Doorbell Product to your telegram account:'  + '\n/link <ProductKey>\n\n'  + 'On Lights:\n/onlight\n\n'  + 'Off Lights:\n/offlight\n\n'  + 'Display Message: (32 char limit)\n/displaymessage <message>'  + '\n\nClear Message:\n/clearmessage'  + '\n\nTake Photo:\n/picture'  + '\n\nTake a 8s Video:\n/video', {});  } else if (userMsg.match(/\/picture|Take Picture/i)) { // publish to take a picture  user.getProductKey(chatId).then(function (productKey) {  var data = {  "device": "camera",  "productKey": productKey,  "action": "take\_picture"  }  device.publish('anedongdong/' + productKey + "/camera", JSON.stringify(data));  pubsub.sendMessage(chatId, "Taking photo...", {});  })  } else if (userMsg.match(/\/onlight|On Light/i)) { // publish to on led  user.getProductKey(chatId).then(function (productKey) {  var data = {  "device": "led",  "productKey": productKey,  "action": "on\_led"  }  device.publish('anedongdong/' + productKey + "/led", JSON.stringify(data));  pubsub.sendMessage(chatId, "Turning On Light..", {});  })  } else if (userMsg.match(/\/offlight|Off Light/i)) { // publish to off led  user.getProductKey(chatId).then(function (productKey) {  var data = {  "device": "led",  "productKey": productKey,  "action": "off\_led"  }  device.publish('anedongdong/' + productKey + "/led", JSON.stringify(data));  pubsub.sendMessage(chatId, "Turning Off Light..", {});  })  } else if (userMsg.match(/\/displaymessage (.+)|Display a Message|\/displaymessage|display message/i)) { // publish to display message  var message = userMsg.match(/\/displaymessage (.+)|Display a Message|\/displaymessage|display message/i)[1];  if (message) {  user.getProductKey(chatId).then(function (productKey) {  var datetime = Math.round((new Date()).getTime() / 1000)  var data = {  "device": "lcd",  "productKey": productKey,  "action": "display\_message",  "data": {  "message": message,  "datetime": datetime  }  }  device.publish('anedongdong/' + productKey + "/lcd", JSON.stringify(data));  pubsub.sendMessage(chatId, "Displaying Message", {});  })  } else {  pubsub.sendMessage(msg.chat.id, "Ok! Please key in the display message in this format: (32 Char limit) /displaymessage <Your Message>", {});  }  } else if (userMsg.match(/\/dosomething/i)) { // display list of things the user can do  let replyOptions = {  reply\_markup: {  resize\_keyboard: true,  one\_time\_keyboard: true,  keyboard: [  ['Take Picture'],  ['Take Video'],  ['Display a Message'],  ['Clear Message'],  ['On Light'],  ["Off Light"]  ],  },  };  pubsub.sendMessage(chatId, "What do you want to do?", replyOptions);  } else if (userMsg.match(/\/video|Take Video/i)) { // publish to take a video  user.getProductKey(chatId).then(function (productKey) {  var data = {  "device": "camera",  "productKey": productKey,  "action": "take\_video"  }  device.publish('anedongdong/' + productKey + "/camera", JSON.stringify(data));  pubsub.sendMessage(chatId, "Taking Video...", {});  })  } else if (userMsg.match(/\/clearmessage|Clear Message/i)) { // publish to clear lcd message  user.getProductKey(chatId).then(function (productKey) {  var data = {  "device": "lcd",  "productKey": productKey,  "action": "clear\_message"  }  device.publish('anedongdong/' + productKey + "/lcd", JSON.stringify(data));  pubsub.sendMessage(chatId, "Clearing Message", {});  })  } else {  pubsub.sendMessage(chatId, 'Try these instead\n\n'  + 'On Lights:\n/onlight\n\n'  + 'Off Lights:\n/offlight\n\n'  + 'Display Message: (32 char limit)\n/displaymessage <message>'  + '\n\nClear Message:\n/clearmessage'  + '\n\nTake Photo:\n/picture'  + '\n\nTake a 8s Video:\n/video', {});  }  } else {  pubsub.sendMessage(chatId, 'Your Telegram account is not linked to any of our products\n /link <ProductKey> to link your account', {});  }  })  }  });  var device = awsIot.device({  keyPath: './certs/private.pem.key',  certPath: './certs/certificate.pem.crt',  caPath: './certs/rootca.pem',  clientId: "anedingdong",  host: config.host  });  device.on('connect', function () {  console.log('MQTT connected');  device.subscribe('anedingding/#');  });  device.on('message', function (topic, payload) {  var payloadData = JSON.parse(payload)  var productKey = payloadData.productKey;  var action = payloadData.action;  user.getTelegramId(productKey).then(function (chatId) {  switch (action) {  case "button\_pressed": { //send telegram message and save to firebase  var datetime = payloadData.data.datetime;  user.buttonPressed(datetime, productKey).then(function () {  let replyOptions = {  reply\_markup: {  resize\_keyboard: true,  one\_time\_keyboard: true,  keyboard: [  ['Take Video'],  ['Display a Message']  ],  },  };  pubsub.sendMessage(chatId, 'Someone has pressed your doorbell, what do you want to do?', replyOptions);  })  break;  }  case "led\_on": {  user.updateLEDStatus("on", productKey).then(function () { })  pubsub.sendMessage(chatId, 'Lights are On', {});  break;  }  case "led\_off": {  user.updateLEDStatus("off", productKey).then(function () { })  pubsub.sendMessage(chatId, 'Lights are Off', {});  break;  }  case "message\_display": {  var datetime = payloadData.data.datetime;  var message = payloadData.data.message;  user.displayMessage(datetime, message, productKey).then(function () { })  pubsub.sendMessage(chatId, 'Message Displayed', {});  break;  }  case "picture\_taken": {  var picturePath = payloadData.data.picturePath;  var datetime = payloadData.data.datetime;  user.pictureTaken(datetime, picturePath, productKey).then(function () {  telegram.sendPhoto(chatId, picturePath, {  caption: "Photo taken"  }).catch(function (error) {  console.log(error)  });  })  break;  }  case "video\_taken": {  var videoPath = payloadData.data.videoPath;  var datetime = payloadData.data.datetime;  user.videoTaken(datetime, videoPath, productKey).then(function () {  telegram.sendDocument(chatId, videoPath, {  caption: "Video taken"  }).catch(function (error) {  console.log(error)  });  })  break;  }  case "message\_clear": {  pubsub.sendMessage(chatId, 'Message Cleared', {});  break;  }  }  })  }); | |
| 1. Create a javascript file **user.js** and paste the code inside the file   This file contains the methods related to Fireabse users node   |  | | --- | | var user = module.exports = {};  var path = require('path');  var firebase = require('firebase');  var moment = require('moment');  var product = require(path.resolve('./APIs/product.js'));  var pubsub = require(path.resolve('./APIs/pubsub.js'));  var rootRef = firebase.database().ref('aneSmartDoorbell');  var userRef = rootRef.child('users');  // Create an account for the user  user.signup = function (useremail, userpassword, productKey) {  return new Promise(function (resolve, reject) {  product.checkProductKey(productKey) //check if product available  .then(function (available) {  if (available == true) {  firebase.auth().createUserWithEmailAndPassword(useremail, userpassword)  .then(function (result) { //uid, email, emailVerified  result.sendEmailVerification()  .then(function () { //send email to user  var newUser = {  "email": result.email,  "productKey": productKey,  "uuid": result.uid,  "telegramChatId": ""  }  var userKey = userRef.push(newUser)  .then(function (data) {  product.linkProductToUser(productKey, data.key);  resolve("Successfully Sign up");  }).catch(function (error) {  console.log("Error: " + error)  reject("Unable to add user");  })  }).catch(function (error) { //verification email sent fail  reject("Verification Email not sent");  });  }).catch(function (error) {  console.log(JSON.stringify(error))  reject(error.message);  });  } else {  reject(available);  }  });  });  } //end of signup()  user.signin = function (useremail, userpassword) {  return new Promise(function (resolve, reject) {  firebase.auth().signInWithEmailAndPassword(useremail, userpassword)  .then(function (result) {  if (result) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().email == useremail) {  resolve({ message: "Successfully Signed in", userid: oneUser.key, productkey: oneUser.val().productKey })  }  })  })  }  }).catch(function (error) {  reject(error.message);  });  })  }  // get telegram id of product  user.getTelegramId = function (productKey) {  return new Promise(function (resolve, reject) {  var gotId = false;  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  resolve(oneUser.val().telegramChatId);  gotId = true;  }  })  if (!gotId) { reject; }  })  })  }  // update the user telegram id  user.updateTelegramId = function (productKey, telegramId) {  return new Promise(function (resolve, reject) {  var updated = false;  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  userRef.child(oneUser.key).child("telegramChatId").set(telegramId);  pubsub.sendMessage(telegramId, "Successfully link Telegram Account to your Product", {});  updated = true;  }  })  updated ? resolve("Success") : resolve("Failed");  })  })  }  // check telegram id link to any product  user.checkTelegramId = function (telegramId) {  return new Promise(function (resolve, reject) {  var isLink = false;  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().telegramChatId == telegramId) {  isLink = true;  resolve(isLink);  }  })  if (!isLink) { resolve(isLink); }  })  })  }  // get telegram id of product  user.getProductKey = function (chatId) {  return new Promise(function (resolve, reject) {  var gotId = false;  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().telegramChatId == chatId) {  resolve(oneUser.val().productKey);  gotId = true;  }  })  if (!gotId) { reject; }  })  })  }  // add record of button press event to firebase  user.buttonPressed = function (datetime, productKey) {  return new Promise(function (resolve, reject) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  var newRecord = {  "datetime": datetime  }  userRef.child(oneUser.key).child("device").child("button").push(newRecord, function (error) {  if (error) {  console.log("Error: " + error)  reject();  } else {  resolve();  }  });  }  })  })  })  }  // add record of picture taken event to firebase  user.pictureTaken = function (datetime, picturePath, productKey) {  return new Promise(function (resolve, reject) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  var newRecord = {  "datetime": datetime,  "picturePath": picturePath  }  userRef.child(oneUser.key).child("device").child("camera").child("picture").push(newRecord, function (error) {  if (error) {  console.log("Error: " + error)  reject();  } else {  resolve();  }  });  }  })  })  })  }  // add record of picture taken event to firebase  user.videoTaken = function (datetime, videoPath, productKey) {  return new Promise(function (resolve, reject) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  var newRecord = {  "datetime": datetime,  "videoPath": videoPath  }  userRef.child(oneUser.key).child("device").child("camera").child("video").push(newRecord, function (error) {  if (error) {  console.log("Error: " + error)  reject();  } else {  resolve();  }  });  }  })  })  })  }  // add record of display message event to firebase  user.displayMessage = function (datetime, message, productKey) {  return new Promise(function (resolve, reject) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  var newRecord = {  "datetime": datetime,  "message": message  }  userRef.child(oneUser.key).child("device").child("lcd").push(newRecord, function (error) {  if (error) {  console.log("Error: " + error)  reject();  } else {  resolve();  }  });  }  })  })  })  }  // update the led status  user.updateLEDStatus = function (status, productKey) {  return new Promise(function (resolve, reject) {  userRef.once('value', function (snapshot) {  snapshot.forEach(function (oneUser) {  if (oneUser.val().productKey == productKey) {  userRef.child(oneUser.key).child("device").child("led").child("status").set(status);  resolve("Updated LED status")  }  })  })  })  }  // return list of past user photos  user.getUserPhoto = function (userKey) {  return new Promise(function (resolve, reject) {  userRef.child(userKey).child("device").child("camera").child("picture").once('value', function (snapshot) {  var photoList = [];  snapshot.forEach(function (data) {  var onePhoto = {  "datetime": moment.unix(data.val().datetime).format("dddd, DD MMMM YYYY, h:mm:ss a"),  "picturePath": data.val().picturePath  }  photoList.push(onePhoto);  });  resolve(photoList)  })  })  }  // return list of past user video downloadable links  user.getUserVideo = function (userKey) {  return new Promise(function (resolve, reject) {  userRef.child(userKey).child("device").child("camera").child("video").once('value', function (snapshot) {  var videoList = [];  snapshot.forEach(function (data) {  var oneVideo = {  "datetime": moment.unix(data.val().datetime).format("dddd, DD MMMM YYYY, h:mm:ss a"),  "videoPath": data.val().videoPath  }  videoList.push(oneVideo);  });  resolve(videoList)  })  })  }  // return list of user button press history  user.getUserButton = function (userKey) {  return new Promise(function (resolve, reject) {  userRef.child(userKey).child("device").child("button").once('value', function (snapshot) {  var buttonList = [];  var currentDate, newGrp;  var existingGrp = false;  var count = 0;  snapshot.forEach(function (data) {  var date = moment.unix(data.val().datetime).format("DD MMM YYYY");  if (currentDate != date) { //different date as before  if (existingGrp) { //second different date  newGrp = {  "date": currentDate,  "count": count  }  buttonList.push(newGrp)  existingGrp = false;  count = 1;  } else { //first different date  currentDate = date;  existingGrp = true;  newGrp = ""  count += 1;  }  } else if (currentDate == date) {  count += 1;  }  });  newGrp = {  "date": currentDate,  "count": count  }  buttonList.push(newGrp)  resolve(buttonList)  })  })  }  // return list of user display message history  user.getUserMessages = function (userKey) {  return new Promise(function (resolve, reject) {  userRef.child(userKey).child("device").child("lcd").once('value', function (snapshot) {  var messageList = [];  snapshot.forEach(function (data) {  var oneMessage = {  "datetime": moment.unix(data.val().datetime).format("dddd, DD MMMM YYYY, h:mm:ss a"),  "message": data.val().message  }  messageList.push(oneMessage);  });  resolve(messageList)  })  })  }  // return status of led  user.getUserLight = function (userKey) {  return new Promise(function (resolve, reject) {  userRef.child(userKey).child("device").child("led").child("status").once('value', function (snapshot) {  resolve(snapshot)  })  })  } | |

|  |
| --- |
| Task |
| 1. Create a javascript file **config.js** in the project main directory  |  | | --- | | .  |  └── config.js | |
| 1. Add the code below into your config.js file  |  | | --- | | var config = module.exports = {}  // AWS  config.aws = {  host: "", // AWS IoT Thing ARN (refer to Figure 1)  port: 8883,  }  // Firebase Credentials  config.firebaseCredentials = {  // firebase console credentials  }  // Telegram Bot Id  config.telegram = {  token: "" // telegram bot token  }  // Gmail Credentials  config.gmail = {  'email': '', // a gmail account  'password': '' // gmail account password  } | |
| Figure 1: |

|  |
| --- |
| Folder Checklist |
| .  |  ├── .git  |  ├── APIs  | ├── email.js  | ├── product.js  | ├── pubsub.js  | └── user.js  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── Controllers  | ├── DeviceAPIController.js  | ├── ProductAPIController.js  | └── UserAPIController.js  |  ├── node\_modules  |  ├── .cfignore  ├── .gitignore  ├── .project  ├── app.js  ├── config.js  ├── LICENSE  ├── manifest.yml  ├── package.json  └── README.md |

Telegram Bot

|  |
| --- |
| Task |
| At the end of this Section, you will have a Telegram Bot created and “linked” to your Node Server |

|  |
| --- |
| Task |
| 1. In Telegram search for @BotFather |
| 1. /newbot to create a Telegram Bot 2. Give your bot a name and a username 3. After the bot is created copy the token as shown below |
| 1. Paste your token in the **config.js** file |
| 1. You can choose to set commands for the bot to help the user when they use the bot |
| For more information of other codes you can do with Telegram Bot: <https://core.telegram.org/bots/api> |

Firebase Console

|  |
| --- |
| Overview |
| At the end of this Section, you will have a Firebase Console app created and “linked” to the Node Server |

|  |
| --- |
| Task |
| 1. Go to <https://console.firebase.google.com/u/0/> and click on “Add project” |
| 1. Enter your Project Name and click Create Project |
| 1. Click on “Add Firebase to your web app” |
| 1. Copy the codes below |
| 1. Paste the code into the config.js file under Firebase Credentials |
| 1. To enable email and password authentication in Firebase, click on Authentication tab |
| 1. At the Sign-in method tab, click on the pencil icon at the right side of the Email/Password |
| 1. Click on Enable and then Save |
| 1. Change the Database rules of the Firebase app to allow the Node Server to read and write to Firebase      |  | | --- | | {  "rules": {  ".read" : true,  ".write" : true  }  } | |

Publish to Bluemix

|  |
| --- |
| Overview |
| In this section, we will publish the updated codes to Bluemix using Git Commands |

|  |
| --- |
| Task |
| 1. Open command prompt in the Node Server project directory |
| 1. Execute the code git add \* to add all the changes in the file |
| 1. git commit -m 'updated codes' to commit the changes added |
| 1. git push -u origin master to push the changes |
| 1. You can check your Bluemix Console to see whether it is updated |

# Section 7 AnE Smart Doorbell Systems (Part 5)

Web Application Setup

|  |
| --- |
| Overview |
| At the end of this section your folder directory should look like this  .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── static  | └── css  | ├── login.css  | ├── register.css  | ├── styles.css  | └── welcome.css  |  ├── templates  | ├── dashboard.html  | ├── lcd.html  | ├── light.html  | ├── login.html  | ├── picture.html  | ├── register.html  | └── welcome.html  |  └── server.py |

### Add AWS Certificates

|  |
| --- |
| Task |
| Add in your AWS Certs into your project directory for the Web application  .  |  └── certs  ├── certificate.pem.crt  ├── private.pem.key  ├── public.pem.key  └── rootca.pem |

### server.py

|  |
| --- |
| Task |
| 1. Create a new file named **server.py** in the project directory |
| 2. Copy the following codes  # coding: utf8  import gevent  import gevent.monkey  from gevent.pywsgi import WSGIServer  from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient  import json  from datetime import datetime  import calendar  gevent.monkey.patch\_all()  from flask import Flask, request, Response, render\_template, jsonify  app = Flask(\_\_name\_\_)  host = "" #specify your REST API Endpoint  rootCAPath = "certs/rootca.pem"  certificatePath = "certs/certificate.pem.crt"  privateKeyPath = "certs/private.pem.key"  my\_rpi = AWSIoTMQTTClient("anesmartdoorbellweb")  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  my\_rpi.connect()  ########### Views ###########  @app.route("/")  def welcome():  return render\_template('welcome.html')  @app.route("/login")  def login():  return render\_template('login.html')  @app.route("/register")  def register():  return render\_template('register.html')  @app.route("/dashboard")  def dashboard():  templateData = {  'title': 'Dashboard',  'main': 'AnE Smart Doorbell'  }  return render\_template('dashboard.html', \*\*templateData)  @app.route("/picture")  def picture():  templateData = {  'title': 'Pictures',  'main': 'AnE Smart Doorbell'  }  return render\_template('picture.html', \*\*templateData)  @app.route("/light")  def light():  templateData = {  'title': 'Lights',  'main': 'AnE Smart Doorbell'  }  return render\_template('light.html', \*\*templateData)  @app.route("/message")  def message():  templateData = {  'title': 'Display Message',  'main': 'AnE Smart Doorbell'  }  return render\_template('lcd.html', \*\*templateData)  @app.route("/led", methods=["POST"])  def led():  data = json.loads(request.data)  if (data["status"] == "on"):  payload=json.dumps({  "device": "led",  "productKey": data["productKey"],  "action": "on\_led"  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/led", payload, 1)  elif (data["status"] == "off"):  payload=json.dumps({  "device": "led",  "productKey": data["productKey"],  "action": "off\_led"  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/led", payload, 1)  return "done"  @app.route("/camera", methods=["POST"])  def camera():  data = json.loads(request.data)  if (data["action"] == "picture"):  payload=json.dumps({  "device": "camera",  "productKey": data["productKey"],  "action": "take\_picture"  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/camera", payload, 1)  elif (data["action"] == "video"):  payload=json.dumps({  "device": "camera",  "productKey": data["productKey"],  "action": "take\_video"  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/camera", payload, 1)  return "done"  @app.route("/lcd", methods=["POST"])  def lcd():  data = json.loads(request.data)  if (data["action"] == "clear"):  payload=json.dumps({  "device": "lcd",  "productKey": data["productKey"],  "action": "clear\_message"  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/lcd", payload, 1)  elif (data["action"] == "display"):  d = datetime.utcnow()  timestamp = calendar.timegm(d.utctimetuple())  payload=json.dumps({  "device": "lcd",  "productKey": data["productKey"],  "action": "display\_message",  "data": {  "message": data["message"],  "datetime": timestamp  }  })  my\_rpi.publish("anedongdong/" + data["productKey"] + "/lcd", payload, 1)  return "done"  if \_\_name\_\_ == '\_\_main\_\_':  try:  http\_server = WSGIServer(('0.0.0.0', 8001), app)  app.debug = True  print("Web page running on port 8001")  http\_server.serve\_forever()  except:  print("Exception") |

|  |
| --- |
| Task |
| 1. Create a new folder named **templates** to hold the html files in your project diectory  .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├──templates  |  └── server.py |

### welcome.html

|  |
| --- |
| Task |
| 1. Create file the file **welcome.html** in the templates folder and copy the following codes:   <!DOCTYPE html>  <html lang="en">  <head>  <!-- Theme Made By www.w3schools.com - No Copyright -->  <title>Bootstrap Theme The Band</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet" type="text/css">  <link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet" type="text/css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  <link rel="stylesheet" href="{{ url\_for('static', filename='css/welcome.css') }}">  <style>    </style>  </head>  <body id="myPage" data-spy="scroll" data-target=".navbar" data-offset="50">  <nav class="navbar navbar-default navbar-fixed-top">  <div class="container-fluid">  <div class="navbar-header">  <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">  <span class="icon-bar"></span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  </button>  <a class="navbar-brand" href="/">AnE Smart Door Bell </a>  </div>  <div class="collapse navbar-collapse" id="myNavbar">  <ul class="nav navbar-nav navbar-right">  <li><a href="login">LOGIN</a></li>  <li><a href="register">REGISTER</a></li>  </ul>  </div>  </div>  </nav>  <div id="myCarousel" class="carousel slide" data-ride="carousel">  <!-- Container (The Band Section) -->  <div id="band" class="container text-center">  <h3>AnE SMART DOOR BELL SYSTEM</h3>  <p><em>Notify you when someone press your doorbell!</em></p>  <p>Telegram Bot</p>  <br>  </div>  <!-- Container (Contact Section) -->  <!-- Add Google Maps -->  <div id="googleMap"></div>  <script>  function myMap() {  var myCenter = new google.maps.LatLng(1.3083, 103.7797);  var mapProp = { center: myCenter, zoom: 16, scrollwheel: false, draggable: false, mapTypeId: google.maps.MapTypeId.ROADMAP };  var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);  var marker = new google.maps.Marker({ position: myCenter });  marker.setMap(map);  var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);  }  </script>    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCbkMdOajBjNodVhXnGW\_LZ874WqIY54ws&callback=myMap"></script>  <!-- Footer -->  <footer class="text-center">  <a class="up-arrow" href="#myPage" data-toggle="tooltip" title="TO TOP">  <span class="glyphicon glyphicon-chevron-up"></span>  </a>  <br><br>  <p>IoT Assignment CA2</p>  </footer>  <script>  $(document).ready(function () {  // Initialize Tooltip  $('[data-toggle="tooltip"]').tooltip();  // Add smooth scrolling to all links in navbar + footer link  $(".navbar a, footer a[href='#myPage']").on('click', function (event) {  // Make sure this.hash has a value before overriding default behavior  if (this.hash !== "") {  // Prevent default anchor click behavior  event.preventDefault();  // Store hash  var hash = this.hash;  // Using jQuery's animate() method to add smooth page scroll  // The optional number (900) specifies the number of milliseconds it takes to scroll to the specified area  $('html, body').animate({  scrollTop: $(hash).offset().top  }, 900, function () {  // Add hash (#) to URL when done scrolling (default click behavior)  window.location.hash = hash;  });  } // End if  });  })  </script>  </body>  </html> |

### login.html

|  |
| --- |
| Task |
| 1. Create file the file **login.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <!-- All the files that are required -->  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ"  crossorigin="anonymous">  <link href='http://fonts.googleapis.com/css?family=Varela+Round' rel='stylesheet' type='text/css'>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/login.css') }}">  <!-- Main Form -->  <div class="login-form-1">  <div class="logo">Login</div>  <form id="login-form" class="text-left">  <div class="login-form-main-message"></div>  <div class="main-login-form">  <div class="login-group">  <div class="form-group">  <label for="txtEmail" class="sr-only">Username</label>  <input type="text" class="form-control" id="txtEmail" name="txtEmail" placeholder="Email">  </div>  <div class="form-group">  <label for="txtPassword" class="sr-only">Password</label>  <input type="password" class="form-control" id="txtPassword" name="txtPassword" placeholder="Password">  </div>  </div>  <button id="btnLogin" class=".login-button btn btn-primary" type="button">Login</i></button>  </div>  <div class="etc-login-form">  <p>Don't have an account? <a href="register">Register</a></p>  </div>  </form>  </div>  <script type="text/javascript">  (function () {  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net";  var userKey = localStorage.getItem("userKey")  var productKey = localStorage.getItem("productKey")  if (userKey != null && productKey != null) {  window.location.href = "dashboard"  }  $('#btnLogin').click(function () {  var emailInput = $('#txtEmail').val()  var passwordInput = $('#txtPassword').val()  swal({  title: 'Logging in',  showConfirmButton: false,  imageUrl: "https://media.tenor.com/images/d6cd5151c04765d1992edfde14483068/tenor.gif",  imageWidth: 80,  imageHeight: 80  });  $.ajax({  url: baseUrl + '/api/user/signin',  method: 'POST',  data: {  "email": emailInput,  "password": passwordInput  },  success: function (response) {  if (response.success) {  localStorage.setItem("userKey", response.userKey)  localStorage.setItem("productKey", response.productKey)  swal({  title: "Success",  text: response.message,  confirmButtonText: "Ok",  type: "success"  }).then(function () {  window.location.href = "dashboard";  });  } else {  $('#txtEmail').val(null)  $('#txtPassword').val(null)  swal("Error", response.message, "error");  }  },  error: function (error) {  console.dir(error)  }  });  });  })();  </script>  <style>  body {  background: #3b3b3b;  padding: 10px;  font-family: 'Varela Round';  }  </style> |

### register.html

|  |
| --- |
| Task |
| 1. Create **register.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <!-- All the files that are required -->  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ"  crossorigin="anonymous">  <link href='http://fonts.googleapis.com/css?family=Varela+Round' rel='stylesheet' type='text/css'>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/register.css') }}">  <!-- Main Form -->  <div class="login-form-1">  <div class="logo">Register</div>  <form id="login-form" class="text-left">  <div class="login-form-main-message"></div>  <div class="main-login-form">  <div class="login-group">  <div class="form-group">  <label for="txtEmail" class="sr-only">Username</label>  <input type="text" class="form-control" id="txtEmail" name="txtEmail" placeholder="Email">  </div>  <div class="form-group">  <label for="txtPassword" class="sr-only">Password</label>  <input type="password" class="form-control" id="txtPassword" name="txtPassword" placeholder="Password">  </div>  <div class="form-group">  <label for="txtProductKey" class="sr-only">Product Key</label>  <input type="text" class="form-control" id="txtProductKey" name="txtProductKey" placeholder="Product Key">  </div>  </div>  <button id="btnRegister" class=".login-button btn btn-primary" type="button">Register</i></button>  </div>  <div class="etc-login-form">  <p>Already have an account? <a href="login">Login</a></p>  </div>  </form>  </div>  <script type="text/javascript">  (function () {  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net";  var userKey = localStorage.getItem("userKey")  var productKey = localStorage.getItem("productKey")  if (userKey != null && productKey != null) {  window.location.href = "dashboard"  }  $('#btnRegister').click(function () {  console.log("ahah")  var emailInput = $('#txtEmail').val()  var passwordInput = $('#txtPassword').val()  var productKeyInput = $('#txtProductKey').val()  $.ajax({  url: baseUrl + '/api/user/signup',  method: 'POST',  data: {  "email": emailInput,  "password": passwordInput,  "productKey": productKeyInput  },  success: function (response) {  if (response.success) {  swal("Success", response.message, "success");  window.location.href = "login"  } else {  swal("Error", response.message, "error");  }  },  error: function (error) {  console.dir(error)  }  });  });  })();  </script> |

### dashboard.html

|  |
| --- |
| Task |
| 1. Create **dashboard.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <head>  <title>{{ title }}</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://gitcdn.github.io/bootstrap-toggle/2.2.2/js/bootstrap-toggle.min.js"></script>  <script src="http://d3js.org/d3.v3.min.js"></script>  <script src="http://dimplejs.org/dist/dimple.v2.1.6.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <link href="https://gitcdn.github.io/bootstrap-toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  </head>  <body>  <div class="container-fluid">  <div class="row content">  <div class="col-sm-3 sidenav hidden-xs">  <h3>{{main}}</h3>  <p> Welcome to AnE smart systems... </p>  <hr/>  <div>  <h3 id="Time"></h3>  <h4 id="Date"></h4>  </div>  <hr />  <ul class="nav navbar-collapse nav-stacked" id="navigations">  <li class="active"><a href="dashboard"><span class="glyphicon glyphicon-home"></span> Dashboard</a></li>  <li><a href="picture"><span class="glyphicon glyphicon-picture"></span> View Pictures</a></li>  <li><a href="light"><span class="glyphicon glyphicon-lamp"></span> Lights</a></li>  <li><a href="message"><span class="glyphicon glyphicon-comment"></span> Messages</a></li>  <li><a href="/" id="logout"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>  </ul>  <br>  </div>  <br>  <div class="col-sm-9">  <h2 class="text-center">{{title}}</h2>  <hr />  <div class="panel panel-default">  <div class="panel-heading custom\_class">  <h3 class="panel-title">  Number of times doorbell is pressed by Days:  </h3>  </div>  <div class="panel-body">  <div id="chartContainer"></div>  </div>  <div class="panel-footer">  <span class="glyphicon glyphicon-time"></span>  <p id="lastUpdate"></p>  </div>  </div>  </div>  </div>  </div>  </body>  <script>  (function () {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net/"  var userKey = localStorage.getItem("userKey")  var svg  var chart  var productKey = localStorage.getItem('productKey')  if (userKey == null || productKey == null) {  swal({  title: "Error",  text: "You are not Login",  confirmButtonText: "Login here",  type: "error"  }).then(function () {  localStorage.removeItem('userKey')  localStorage.removeItem('productKey')  window.location.href = "login";  });  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //###############################################  //--------------- checkTime()------------------//  // add zero in front of numbers < 10  //###############################################  function checkTime(i) {  if (i < 10) { i = "0" + i };  return i;  }  //###############################################  //--------------- startTime()------------------//  // start running the clock on the web page  //###############################################  function startTime() {  var now = new Date;  var map = { '13': 1, '14': 2, '15': 3, '16': 4, '17': 5, '18': 6, '19': 7, '20': 8, '21': 9, '22': 10, '23': 11, '00': 12 }  h = checkTime(now.getHours());  m = checkTime(now.getMinutes());  s = checkTime(now.getSeconds());  var ampm = (h >= 12) ? "PM" : "AM";  var hh = (h > 12 || h == 00) ? map[h] : h  var time = hh + ':' + m + ':' + s + ' ' + ampm;  document.getElementById('Time').innerHTML = time  }  //##########################################  //--------------- date()------------------//  // display the date on the web page  //##########################################  function date() {  var now = new Date;  year = now.getFullYear();  month = now.getMonth();  months = new Array('January', 'February', 'March', 'April', 'May', 'June', 'Jully', 'August', 'September', 'October', 'November', 'December');  d = now.getDate();  day = now.getDay();  days = new Array('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday');  date = '' + days[day] + ', ' + d + ' ' + months[month] + ' ' + year;  $('#Date').text(date)  }  //##########################################  //--------------- time()------------------//  // returns formatted current time  //##########################################  function time() {  var now = new Date;  var map = { '13': 1, '14': 2, '15': 3, '16': 4, '17': 5, '18': 6, '19': 7, '20': 8, '21': 9, '22': 10, '23': 11, '00': 12 }  h = checkTime(now.getHours());  m = checkTime(now.getMinutes());  s = checkTime(now.getSeconds());  var ampm = (h >= 12) ? "PM" : "AM";  var hh = (h > 12 || h == 00) ? map[h] : h  var time = hh + ':' + m + ':' + s + ' ' + ampm;  return time;  }  //#####################################################  //--------------- onRefreshChart()------------------//  // get latest light sensor data  //#####################################################  function onRefreshChart() {  var now = new Date();  $('#lastUpdate').text("Last Updated: " + now.getDate() + "/" + (now.getMonth() + 1) + "/" + now.getFullYear() + " " + time())  $.ajax({  url: baseUrl + "api/device/button",  method: "GET",  headers: {  "content-type": "application/json",  "userkey": userKey  },  success: function (result) {  onGetResult(result.data)  }  })  }  //##################################################  //--------------- onGetResult()------------------//  // loads data into chart  //##################################################  function onGetResult(result) {  chart.data = result;  chart.draw(500);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Run Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  date();  setInterval(startTime, 500);  svg = dimple.newSvg("#chartContainer", 800, 400);  chart = new dimple.chart(svg, null);  x = chart.addCategoryAxis("x", "date");  x.title = "Date";  y = chart.addMeasureAxis("y", "count");  y.title = "Number of times pressed";  series = chart.addSeries(null, dimple.plot.bar);  onRefreshChart();  setInterval(onRefreshChart, 10000);  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Events \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  $('#logout').on('click', function () {  swal({  title: 'Logging out',  showConfirmButton: false,  });  localStorage.removeItem("userKey");  localStorage.removeItem("productKey");  })  }())  </script> |

### lcd.html

|  |
| --- |
| Task |
| 1. Create **lcd.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <head>  <title>{{ title }}</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://gitcdn.github.io/bootstrap-toggle/2.2.2/js/bootstrap-toggle.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <link href="https://gitcdn.github.io/bootstrap-toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  </head>  <body>  <div class="container-fluid">  <div class="row content">  <div class="col-sm-3 sidenav hidden-xs">  <h3>{{main}}</h3>  <p> Welcome to AnE smart systems... </p>  <hr />  <div>  <h3 id="Time"></h3>  <h4 id="Date"></h4>  </div>  <hr />  <ul class="nav navbar-collapse nav-stacked" id="navigations">  <li class="active"><a href="dashboard"><span class="glyphicon glyphicon-home"></span> Dashboard</a></li>  <li><a href="picture"><span class="glyphicon glyphicon-picture"></span> View Pictures</a></li>  <li><a href="light"><span class="glyphicon glyphicon-lamp"></span> Lights</a></li>  <li><a href="message"><span class="glyphicon glyphicon-comment"></span> Messages</a></li>  <li><a href="/" id="logout"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>  </ul>  <br>  </div>  <br>  <div class="col-sm-9">  <h2 class="text-center">{{title}}</h2>  <hr />  <div class="panel panel-default">  <div class="panel-heading custom\_class">  <h3 class="panel-title">  Camera  </h3>  </div>  <div class="panel-body">  <label for="messageInput">Display Message:</label>  <input type="text" class="form-control" id="messageInput" maxlength="32"><br />  <button id="displayMessage" class="btn primary">Display Message on LCD</button>  <button id="clearMessage" class="btn primary">Clear Message on LCD</button>  </div>  </div>  <div class="panel panel-default">  <div class="panel-heading custom\_class">  <h3 class="panel-title">  Message History  </h3>  </div>  <div class="panel-body">  <table id="userMessage"></table>  </div>  <div class="panel-footer">  </div>  </div>  </div>  </div>  </div>  </body>  <script>  (function () {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net";  var userKey = localStorage.getItem('userKey')  var productKey = localStorage.getItem('productKey')  if (userKey == null || productKey == null) {  swal({  title: "Error",  text: "You are not Login",  confirmButtonText: "Login here",  type: "error"  }).then(function () {  localStorage.removeItem('userKey')  localStorage.removeItem('productKey')  window.location.href = "login";  });  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  function displayMsgHistory() {  $.ajax({  // Post username, password & the grant type to /token  url: baseUrl + '/api/device/message',  method: 'GET',  headers: {  "content-type": "application/json",  "userkey": userKey  },  success: function (response) {  messageList = response.data;  var $tableElement = $('#userMessage');  var $dateTimeHeaderElement = $('<th></th>');  $dateTimeHeaderElement.addClass('col-md-4 col-sm-6')  $dateTimeHeaderElement.text("DateTimeSent");  var $messageHeaderElement = $('<th></th>');  $messageHeaderElement.addClass('col-md-4 col-sm-6')  $messageHeaderElement.text("Message");  var $headerRowElement = $('<tr></tr>');  $headerRowElement.append($dateTimeHeaderElement);  $headerRowElement.append($messageHeaderElement);  $tableElement.append($headerRowElement);  messageList.forEach(function (oneMessage) {  var messageDateTime = oneMessage.datetime  var messageMsg = oneMessage.message;  var $oneMessageRowElement = $('<tr></tr>');  var $dateTimeElement = $('<td></td>');  $dateTimeElement.addClass('col-md-4 col-sm-6')  $dateTimeElement.text(messageDateTime);  var $messageElement = $('<td></td>');  $messageElement.addClass('col-md-4 col-sm-6')  $messageElement.text(messageMsg);  $oneMessageRowElement.append($dateTimeElement);  $oneMessageRowElement.append($messageElement);  $tableElement.append($oneMessageRowElement)  });  },  error: function (error) {  console.dir(error)  }  });  }  //###############################################  //--------------- checkTime()------------------//  // add zero in front of numbers < 10  //###############################################  function checkTime(i) {  if (i < 10) { i = "0" + i };  return i;  }  //###############################################  //--------------- startTime()------------------//  // start running the clock on the web page  //###############################################  function startTime() {  var now = new Date;  var map = { '13': 1, '14': 2, '15': 3, '16': 4, '17': 5, '18': 6, '19': 7, '20': 8, '21': 9, '22': 10, '23': 11, '00': 12 }  h = checkTime(now.getHours());  m = checkTime(now.getMinutes());  s = checkTime(now.getSeconds());  var ampm = (h >= 12) ? "PM" : "AM";  var hh = (h > 12 || h == 00) ? map[h] : h  var time = hh + ':' + m + ':' + s + ' ' + ampm;  document.getElementById('Time').innerHTML = time  var t = setTimeout(startTime, 500);  }  //##########################################  //--------------- date()------------------//  // display the date on the web page  //##########################################  function date() {  var now = new Date;  year = now.getFullYear();  month = now.getMonth();  months = new Array('January', 'February', 'March', 'April', 'May', 'June', 'Jully', 'August', 'September', 'October', 'November', 'December');  d = now.getDate();  day = now.getDay();  days = new Array('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday');  date = '' + days[day] + ', ' + d + ' ' + months[month] + ' ' + year;  $('#Date').text(date)  }  $('#displayMessage').on('click', function () {  $.ajax({  url: "/lcd",  type: "POST",  contentType: "application/json",  data: JSON.stringify({  "action": "display",  "productKey": productKey,  "message": $('#messageInput').val()  }),  success: function (response) { }  })  })  $('#clearMessage').on('click', function () {  $.ajax({  url: "/lcd",  type: "POST",  contentType: "application/json",  data: JSON.stringify({  "action": "clear",  "productKey": productKey  }),  success: function (response) { }  })  })  $('#logout').on('click', function () {  swal({  title: 'Logging out',  showConfirmButton: false,  });  localStorage.removeItem("userKey");  localStorage.removeItem("productKey");  })  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Run Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  date();  setInterval(startTime, 500)  displayMsgHistory()  }())  </script>  <style>  #userMessage {  width: 90%;  margin: auto;  }  table,  th,  td {  border: 1px solid black;  border-collapse: collapse;  }  </style> |

### light.html

|  |
| --- |
| Task |
| 1. Create **light.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <!DOCTYPE html>  <head>  <title>{{ title }}</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://gitcdn.github.io/bootstrap-toggle/2.2.2/js/bootstrap-toggle.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <link href="https://gitcdn.github.io/bootstrap-toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  </head>  <body>  <div class="container-fluid">  <div class="row content">  <div class="col-sm-3 sidenav hidden-xs">  <h3>{{main}}</h3>  <p> Welcome to AnE smart systems... </p>  <hr />  <div>  <h3 id="Time"></h3>  <h4 id="Date"></h4>  </div>  <hr />  <ul class="nav navbar-collapse nav-stacked" id="navigations">  <li><a href="dashboard"><span class="glyphicon glyphicon-home"></span> Dashboard</a></li>  <li><a href="picture"><span class="glyphicon glyphicon-picture"></span> View Pictures</a></li>  <li class="active"><a href="light"><span class="glyphicon glyphicon-lamp"></span> Lights</a></li>  <li><a href="message"><span class="glyphicon glyphicon-comment"></span> Messages</a></li>  <li><a href="/" id="logout"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>  </ul>  <br>  </div>  <br>  <div class="col-sm-9">  <h2 class="text-center">{{title}}</h2>  <hr />  <div class="panel panel-default">  <div class="panel-heading custom\_class">  <h3 class="panel-title">  Light Status  </h3>  </div>  <div class="panel-body">  <section class="row text-center placeholders">  <div class="col-6 col-sm-3 placeholder">  <img src="https://media.licdn.com/mpr/mpr/AAEAAQAAAAAAAAdeAAAAJGZjMTBkYzRkLWYzZTgtNGE1YS1iOWQ2LTk0ZmRhNjZhNGQ1ZQ.png" width="80"  height="80" class="img-fluid rounded-circle" alt="Generic placeholder thumbnail">  <h4>LED</h4>  <span class="text-muted">Lights up when user wants to deter intruders.</span><br/><br/>  <input id="ledStatus" type="checkbox" checked data-toggle="toggle">  </div>  <!--<div class="col-6 col-sm-3 placeholder">-->  <!-- <label id="switch" class="switch">  <input type="checkbox">  <span class="on">ON</span><span class="off">OFF</span>  <span class="slider round"></span>  </label> -->  <!--</div>-->  </section>  </div>  <div class="panel-footer">  </div>  </div>  </div>  </div>  </div>  </body>  </html>  <script>  (function () {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  var ledStatus = $('#ledStatus');;  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net";  var userKey = localStorage.getItem('userKey')  var productKey = localStorage.getItem('productKey')  if (userKey == null || productKey == null) {  swal({  title: "Error",  text: "You are not Login",  confirmButtonText: "Login here",  type: "error"  }).then(function () {  localStorage.removeItem('userKey')  localStorage.removeItem('productKey')  window.location.href = "login";  });  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  function updateToggle() {  $.ajax({  url: baseUrl + '/api/device/led',  method: 'GET',  headers: {  "content-type": "application/json",  "userkey": userKey  },  success: function (response) {  if (response.success) {  var status = response.data == "on" ? true : false  if (status) {  if (status != $('#ledStatus').prop('checked')) { $('#ledStatus').bootstrapToggle('on') }  } else {  if (status != $('#ledStatus').prop('checked')) { $('#ledStatus').bootstrapToggle('off') }  }  }  },  error: function (error) {  console.dir(error)  }  });  }  //###############################################  //--------------- checkTime()------------------//  // add zero in front of numbers < 10  //###############################################  function checkTime(i) {  if (i < 10) { i = "0" + i };  return i;  }  //###############################################  //--------------- startTime()------------------//  // start running the clock on the web page  //###############################################  function startTime() {  var now = new Date;  var map = { '13': 1, '14': 2, '15': 3, '16': 4, '17': 5, '18': 6, '19': 7, '20': 8, '21': 9, '22': 10, '23': 11, '00': 12 }  h = checkTime(now.getHours());  m = checkTime(now.getMinutes());  s = checkTime(now.getSeconds());  var ampm = (h >= 12) ? "PM" : "AM";  var hh = (h > 12 || h == 00) ? map[h] : h  var time = hh + ':' + m + ':' + s + ' ' + ampm;  document.getElementById('Time').innerHTML = time  }  //##########################################  //--------------- date()------------------//  // display the date on the web page  //##########################################  function date() {  var now = new Date;  year = now.getFullYear();  month = now.getMonth();  months = new Array('January', 'February', 'March', 'April', 'May', 'June', 'Jully', 'August', 'September', 'October', 'November', 'December');  d = now.getDate();  day = now.getDay();  days = new Array('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday');  date = '' + days[day] + ', ' + d + ' ' + months[month] + ' ' + year;  $('#Date').text(date)  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Run Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  date();  setInterval(startTime, 500)  setInterval(updateToggle, 10000)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Events \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  var firstTime = true  $('#ledStatus').on("change", function () {  if (!firstTime) {  var status = $(this).prop('checked') ? "on" : "off";  $.ajax({  url: "/led",  type: "POST",  contentType: "application/json",  data: JSON.stringify({  "status": status,  "productKey": productKey  }),  success: function (response) { }  })  } else {  firstTime = false  }  });  $('#logout').on('click', function () {  swal({  title: 'Logging out',  showConfirmButton: false,  });  localStorage.removeItem("userKey");  localStorage.removeItem("productKey");  })  }())  </script>  <style>    .switch {  position: relative;  display: inline-block;  width: 60px;  height: 34px;  }  /\* Hide default HTML checkbox \*/  .switch input {  display: none;  }  /\* The slider \*/  .slider {  position: absolute;  cursor: pointer;  top: 0;  left: 0;  right: 0;  bottom: 0;  background-color: #ccc;  -webkit-transition: .4s;  transition: .4s;  }  .slider:before {  position: absolute;  content: "";  height: 26px;  width: 26px;  left: 4px;  bottom: 4px;  background-color: white;  -webkit-transition: .4s;  transition: .4s;  }  input:checked+.slider {  background-color: #2196F3;  }  input:focus+.slider {  box-shadow: 0 0 1px #2196F3;  }  input:checked+.slider:before {  -webkit-transform: translateX(26px);  -ms-transform: translateX(26px);  transform: translateX(26px);  }  /\* Rounded sliders \*/  .slider.round {  border-radius: 34px;  }  .slider.round:before {  border-radius: 50%;  }  .slider:after {  content: 'OFF';  color: white;  display: block;  position: absolute;  transform: translate(-50%, -50%);  top: 50%;  left: 50%;  font-size: 10px;  font-family: Verdana, sans-serif;  }  input:checked+.slider:after {  content: 'ON';  }  </style> |

### picture.html

|  |
| --- |
| Task |
| 1. Create **pictures.html** in the templates folder and copy the following codes: 2. Replace the baseUrl with your Bluemix Node.js App url   <head>  <title>{{ title }}</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script src="https://gitcdn.github.io/bootstrap-toggle/2.2.2/js/bootstrap-toggle.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.min.js"></script>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/6.6.7/sweetalert2.css" />  <link href="https://gitcdn.github.io/bootstrap-toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">  </head>  <body>  <div class="container-fluid">  <div class="row content">  <div class="col-sm-3 sidenav hidden-xs">  <h3>{{main}}</h3>  <p> Welcome to AnE smart systems... </p>  <hr />  <div>  <h3 id="Time"></h3>  <h4 id="Date"></h4>  </div>  <hr />  <ul class="nav navbar-collapse nav-stacked" id="navigations">  <li><a href="dashboard"><span class="glyphicon glyphicon-home"></span> Dashboard</a></li>  <li class="active"><a href="picture"><span class="glyphicon glyphicon-picture"></span> View Pictures</a></li>  <li><a href="light"><span class="glyphicon glyphicon-lamp"></span> Lights</a></li>  <li><a href="message"><span class="glyphicon glyphicon-comment"></span> Messages</a></li>  <li><a href="/" id="logout"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>  </ul>  <br>  </div>  <br>  <div class="col-sm-9">  <h2 class="text-center">{{title}}</h2>  <hr />  <div class="panel panel-default">  <div class="panel-heading custom\_class">  <h3 class="panel-title">  Camera  </h3>  </div>  <div class="panel-body">  <button id="takePicture" class="btn primary">Take a Picture</button>  <button id="takeVideo" class="btn primary">Take a Video</button>  </div>  </div>  <div class="col-lg-12">  <h3>Pictures Taken</h3>  <hr>  </div>  <div class="row text-center" id="userImg"> </div>  <div class="col-lg-12">  <h3>Videos Taken</h3>  <hr>  </div>  <div class="row text-center" id="userVideo"> </div>  </div>  </div>  </div>  </body>  <script>  (function () {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  var baseUrl = "https://anesmartdoorbell.au-syd.mybluemix.net/"  var userKey = localStorage.getItem("userKey")  var productKey = localStorage.getItem('productKey')  if (userKey == null || productKey == null) {  swal({  title: "Error",  text: "You are not Login",  confirmButtonText: "Login here",  type: "error"  }).then(function () {  localStorage.removeItem('userKey')  localStorage.removeItem('productKey')  window.location.href = "login";  });  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //###############################################  //--------------- checkTime()------------------//  // add zero in front of numbers < 10  //###############################################  function checkTime(i) {  if (i < 10) { i = "0" + i };  return i;  }  //###############################################  //--------------- startTime()------------------//  // start running the clock on the web page  //###############################################  function startTime() {  var now = new Date;  var map = { '13': 1, '14': 2, '15': 3, '16': 4, '17': 5, '18': 6, '19': 7, '20': 8, '21': 9, '22': 10, '23': 11, '00': 12 }  h = checkTime(now.getHours());  m = checkTime(now.getMinutes());  s = checkTime(now.getSeconds());  var ampm = (h >= 12) ? "PM" : "AM";  var hh = (h > 12 || h == 00) ? map[h] : h  var time = hh + ':' + m + ':' + s + ' ' + ampm;  document.getElementById('Time').innerHTML = time  }  //##########################################  //--------------- date()------------------//  // display the date on the web page  //##########################################  function date() {  var now = new Date;  year = now.getFullYear();  month = now.getMonth();  months = new Array('January', 'February', 'March', 'April', 'May', 'June', 'Jully', 'August', 'September', 'October', 'November', 'December');  d = now.getDate();  day = now.getDay();  days = new Array('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday');  date = '' + days[day] + ', ' + d + ' ' + months[month] + ' ' + year;  $('#Date').text(date)  }  //##########################################  //------------ listPicture()---------------//  // display pictures  //##########################################  function listPicture() {  $('#userImg').empty()  $.ajax({  "url": baseUrl + "api/device/photo",  "method": "GET",  "headers": {  "content-type": "application/json",  "userkey": userKey  }, success: function (response) {  imgList = response.data  imgList.forEach(function (oneImg) {  var imgDescription = oneImg.datetime;  var $photoClickLink = $('<a>').attr("href", "#");  var $container = $('<div>').addClass('col-md-4 col-sm-6');  var $imageContainer = $('<div>').addClass('thumbnail');  var $image = $('<img>').attr('src', oneImg.picturePath);  var $captionContainer = $('<div>').addClass('caption');  var $caption = $('<p>').text(imgDescription);  var $containerImg = $imageContainer.append($image);  var $containerCaption = $captionContainer.append($caption);  var $containerThumbnail = $imageContainer.append($containerImg).append($containerCaption);  var $parentContainer = $container.append($containerThumbnail);  var $finalContainer = $photoClickLink.append($parentContainer);  $('#userImg').append($finalContainer);  });  }  });  }  //##########################################  //------------ listVideo()---------------//  // display videos  //##########################################  function listVideo() {  $('#userVideo').empty()  $.ajax({  "url": baseUrl + "api/device/video",  "method": "GET",  "headers": {  "content-type": "application/json",  "userkey": userKey  }, success: function (response) {  response.data.forEach(function (oneVid) {  var vidDescription = oneVid.datetime;  var $container = $('<div>').addClass('col-md-6 col-sm-6 video');  var $video = $('<video width="320" height="240" controls>');  var $videoSrc = $('<source>').attr('src', oneVid.videoPath);  var $videoContainer = $video.append($videoSrc);  var $captionContainer = $('<div>').addClass('caption');  var $caption = $('<p>').text(vidDescription);  var $containerCaption = $captionContainer.append($caption);  var $parentContainer = $container.append($videoContainer).append($containerCaption);  $('#userVideo').append($parentContainer);  });  }  });  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Run Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  date();  listPicture();  listVideo();  setInterval(startTime, 500)  setInterval(listPicture, 60000)  setInterval(listVideo, 60000)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Events \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  $('#takePicture').on('click', function () {  $.ajax({  url: "/camera",  type: "POST",  contentType: "application/json",  data: JSON.stringify({  "action": "picture",  "productKey": productKey  }),  success: function (response) { }  })  })  $('#takeVideo').on('click', function () {  $.ajax({  url: "/camera",  type: "POST",  contentType: "application/json",  data: JSON.stringify({  "action": "video",  "productKey": productKey  }),  success: function (response) { }  })  })  $('#logout').on('click', function () {  swal({  title: 'Logging out',  showConfirmButton: false,  });  localStorage.removeItem("userKey");  localStorage.removeItem("productKey");  })  }())  </script> |

|  |
| --- |
| Setup static folder |
| 1. Create a new folder named **static/css**  .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── static  | └── css  |  ├── templates  | ├── dashboard.html  | ├── lcd.html  | ├── light.html  | ├── login.html  | ├── picture.html  | ├── register.html  | └── welcome.html  |  └── server.py |

### welcome.css

|  |
| --- |
| welcome.css |
| 1. Create **welcome.css** in the static/css folder and copy the following codes:  body {  font: 400 15px/1.8 Lato, sans-serif;  color: #777;  }  h3,  h4 {  margin: 10px 0 30px 0;  letter-spacing: 10px;  font-size: 20px;  color: #111;  }  .container {  padding: 80px 120px;  }  .person {  border: 10px solid transparent;  margin-bottom: 25px;  width: 80%;  height: 80%;  opacity: 0.7;  }  .person:hover {  border-color: #f1f1f1;  }  .carousel-inner img {  -webkit-filter: grayscale(90%);  filter: grayscale(90%);  /\* make all photos black and white \*/  width: 100%;  /\* Set width to 100% \*/  margin: auto;  }  .carousel-caption h3 {  color: #fff !important;  }  @media (max-width: 600px) {  .carousel-caption {  display: none;  /\* Hide the carousel text when the screen is less than 600 pixels wide \*/  }  }  .bg-1 {  background: #2d2d30;  color: #bdbdbd;  }  .bg-1 h3 {  color: #fff;  }  .bg-1 p {  font-style: italic;  }  .list-group-item:first-child {  border-top-right-radius: 0;  border-top-left-radius: 0;  }  .list-group-item:last-child {  border-bottom-right-radius: 0;  border-bottom-left-radius: 0;  }  .thumbnail {  padding: 0 0 15px 0;  border: none;  border-radius: 0;  }  .thumbnail p {  margin-top: 15px;  color: #555;  }  .btn {  padding: 10px 20px;  background-color: #333;  color: #f1f1f1;  border-radius: 0;  transition: .2s;  }  .btn:hover,  .btn:focus {  border: 1px solid #333;  background-color: #fff;  color: #000;  }  .modal-header,  h4,  .close {  background-color: #333;  color: #fff !important;  text-align: center;  font-size: 30px;  }  .modal-header,  .modal-body {  padding: 40px 50px;  }  .nav-tabs li a {  color: #777;  }  #googleMap {  width: 100%;  height: 400px;  -webkit-filter: grayscale(100%);  filter: grayscale(100%);  }  .navbar {  font-family: Montserrat, sans-serif;  margin-bottom: 0;  background-color: #2d2d30;  border: 0;  font-size: 11px !important;  letter-spacing: 4px;  opacity: 0.9;  }  .navbar li a,  .navbar .navbar-brand {  color: #d5d5d5 !important;  }  .navbar-nav li a:hover {  color: #fff !important;  }  .navbar-nav li.active a {  color: #fff !important;  background-color: #29292c !important;  }  .navbar-default .navbar-toggle {  border-color: transparent;  }  .open .dropdown-toggle {  color: #fff;  background-color: #555 !important;  }  .dropdown-menu li a {  color: #000 !important;  }  .dropdown-menu li a:hover {  background-color: #555 !important;  }  footer {  background-color: #2d2d30;  color: #f5f5f5;  padding: 32px;  }  footer a {  color: #f5f5f5;  }  footer a:hover {  color: #777;  text-decoration: none;  }  .form-control {  border-radius: 0;  }  textarea {  resize: none;  } |

### login.css

|  |
| --- |
| login.css |
| 1. Create **login.css** in the static/css folder and copy the following codes:  a {  color: #aaaaaa;  transition: all ease-in-out 200ms;  }  a:hover {  color: #333333;  text-decoration: none;  }  /\*=== 3. Text Outside the Box ===\*/  .etc-login-form {  color: #919191;  padding: 10px 20px;  clear: right;  }  .etc-login-form p {  margin-bottom: 5px;  }  /\*=== 4. Main Form ===\*/  .login-form-1 {  max-width: 300px;  border-radius: 5px;  /\*display: inline-block;\*/  margin: auto;  }  .main-login-form {  position: relative;  }  .login-form-1 .form-control {  border: 0;  box-shadow: 0 0 0;  border-radius: 0;  background: transparent;  color: #555555;  padding: 7px 0;  font-weight: bold;  height: auto;  }  .login-form-1 .form-control::-webkit-input-placeholder {  color: #999999;  }  .login-form-1 .form-control:-moz-placeholder,  .login-form-1 .form-control::-moz-placeholder,  .login-form-1 .form-control:-ms-input-placeholder {  color: #999999;  }  .login-form-1 .form-group {  margin-bottom: 0;  border-bottom: 2px solid #efefef;  padding-right: 20px;  position: relative;  }  .login-form-1 .form-group:last-child {  border-bottom: 0;  }  .login-group {  background: #ffffff;  color: #999999;  border-radius: 8px;  padding: 10px 20px;  margin-bottom: 5%;  }  .login-group-checkbox {  padding: 5px 0;  }  /\*=== 5. Login Button ===\*/  .login-form-1 .login-button {  position: absolute;  right: -25px;  top: 50%;  background: #ffffff;  color: #999999;  padding: 11px 0;  width: 50px;  height: 50px;  margin-top: -25px;  border: 5px solid #efefef;  border-radius: 50%;  transition: all ease-in-out 500ms;  float: right;  }  .login-form-1 .login-button:hover {  color: #555555;  transform: rotate(450deg);  }  .login-form-1 .login-button.clicked {  color: #555555;  }  .login-form-1 .login-button.clicked:hover {  transform: none;  }  .login-form-1 .login-button.clicked.success {  color: #2ecc71;  }  .login-form-1 .login-button.clicked.error {  color: #e74c3c;  }  /\*=== 6. Form Invalid ===\*/  label.form-invalid {  position: absolute;  top: 0;  right: 0;  z-index: 5;  display: block;  margin-top: -25px;  padding: 7px 9px;  background: #777777;  color: #ffffff;  border-radius: 5px;  font-weight: bold;  font-size: 11px;  }  label.form-invalid:after {  top: 100%;  right: 10px;  border: solid transparent;  content: " ";  height: 0;  width: 0;  position: absolute;  pointer-events: none;  border-color: transparent;  border-top-color: #777777;  border-width: 6px;  }  /\*=== 7. Form - Main Message ===\*/  .login-form-main-message {  background: #ffffff;  color: #999999;  border-left: 3px solid transparent;  border-radius: 3px;  margin-bottom: 8px;  font-weight: bold;  height: 0;  padding: 0 20px 0 17px;  opacity: 0;  transition: all ease-in-out 200ms;  }  .login-form-main-message.show {  height: auto;  opacity: 1;  padding: 10px 20px 10px 17px;  }  .login-form-main-message.success {  border-left-color: #2ecc71;  }  .login-form-main-message.error {  border-left-color: #e74c3c;  }  /\*=== 9. Misc ===\*/  .logo {  padding: 15px 0;  font-size: 25px;  color: #aaaaaa;  font-weight: bold;  }  #btnLogin {  float: right;  } |

### register.css

|  |
| --- |
| register.css |
| 1. Create **reigster.css** in the static/css folder and copy the following codes:  a {  color: #aaaaaa;  transition: all ease-in-out 200ms;  }  a:hover {  color: #333333;  text-decoration: none;  }  /\*=== 3. Text Outside the Box ===\*/  .etc-login-form {  color: #919191;  padding: 10px 20px;  clear: right;  }  .etc-login-form p {  margin-bottom: 5px;  }  /\*=== 4. Main Form ===\*/  .login-form-1 {  max-width: 300px;  border-radius: 5px;  /\*display: inline-block;\*/  margin: auto;  }  .main-login-form {  position: relative;  }  .login-form-1 .form-control {  border: 0;  box-shadow: 0 0 0;  border-radius: 0;  background: transparent;  color: #555555;  padding: 7px 0;  font-weight: bold;  height: auto;  }  .login-form-1 .form-control::-webkit-input-placeholder {  color: #999999;  }  .login-form-1 .form-control:-moz-placeholder,  .login-form-1 .form-control::-moz-placeholder,  .login-form-1 .form-control:-ms-input-placeholder {  color: #999999;  }  .login-form-1 .form-group {  margin-bottom: 0;  border-bottom: 2px solid #efefef;  padding-right: 20px;  position: relative;  }  .login-form-1 .form-group:last-child {  border-bottom: 0;  }  .login-group {  background: #ffffff;  color: #999999;  border-radius: 8px;  padding: 10px 20px;  margin-bottom: 5%;  }  .login-group-checkbox {  padding: 5px 0;  }  /\*=== 5. Login Button ===\*/  .login-form-1 .login-button {  position: absolute;  right: -25px;  top: 50%;  background: #ffffff;  color: #999999;  padding: 11px 0;  width: 50px;  height: 50px;  margin-top: -25px;  border: 5px solid #efefef;  border-radius: 50%;  transition: all ease-in-out 500ms;  float: right;  }  .login-form-1 .login-button:hover {  color: #555555;  transform: rotate(450deg);  }  .login-form-1 .login-button.clicked {  color: #555555;  }  .login-form-1 .login-button.clicked:hover {  transform: none;  }  .login-form-1 .login-button.clicked.success {  color: #2ecc71;  }  .login-form-1 .login-button.clicked.error {  color: #e74c3c;  }  /\*=== 6. Form Invalid ===\*/  label.form-invalid {  position: absolute;  top: 0;  right: 0;  z-index: 5;  display: block;  margin-top: -25px;  padding: 7px 9px;  background: #777777;  color: #ffffff;  border-radius: 5px;  font-weight: bold;  font-size: 11px;  }  label.form-invalid:after {  top: 100%;  right: 10px;  border: solid transparent;  content: " ";  height: 0;  width: 0;  position: absolute;  pointer-events: none;  border-color: transparent;  border-top-color: #777777;  border-width: 6px;  }  /\*=== 7. Form - Main Message ===\*/  .login-form-main-message {  background: #ffffff;  color: #999999;  border-left: 3px solid transparent;  border-radius: 3px;  margin-bottom: 8px;  font-weight: bold;  height: 0;  padding: 0 20px 0 17px;  opacity: 0;  transition: all ease-in-out 200ms;  }  .login-form-main-message.show {  height: auto;  opacity: 1;  padding: 10px 20px 10px 17px;  }  .login-form-main-message.success {  border-left-color: #2ecc71;  }  .login-form-main-message.error {  border-left-color: #e74c3c;  }  /\*=== 8. Custom Checkbox & Radio ===\*/  /\* Base for label styling \*/  [type="checkbox"]:not(:checked),  [type="checkbox"]:checked,  [type="radio"]:not(:checked),  [type="radio"]:checked {  position: absolute;  left: -9999px;  }  [type="checkbox"]:not(:checked)+label,  [type="checkbox"]:checked+label,  [type="radio"]:not(:checked)+label,  [type="radio"]:checked+label {  position: relative;  padding-left: 25px;  padding-top: 1px;  cursor: pointer;  }  /\* checkbox aspect \*/  [type="checkbox"]:not(:checked)+label:before,  [type="checkbox"]:checked+label:before,  [type="radio"]:not(:checked)+label:before,  [type="radio"]:checked+label:before {  content: '';  position: absolute;  left: 0;  top: 2px;  width: 17px;  height: 17px;  border: 0px solid #aaa;  background: #f0f0f0;  border-radius: 3px;  box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3);  }  /\* checked mark aspect \*/  [type="checkbox"]:not(:checked)+label:after,  [type="checkbox"]:checked+label:after,  [type="radio"]:not(:checked)+label:after,  [type="radio"]:checked+label:after {  position: absolute;  color: #555555;  transition: all .2s;  }  /\* checked mark aspect changes \*/  [type="checkbox"]:not(:checked)+label:after,  [type="radio"]:not(:checked)+label:after {  opacity: 0;  transform: scale(0);  }  [type="checkbox"]:checked+label:after,  [type="radio"]:checked+label:after {  opacity: 1;  transform: scale(1);  }  /\* disabled checkbox \*/  [type="checkbox"]:disabled:not(:checked)+label:before,  [type="checkbox"]:disabled:checked+label:before,  [type="radio"]:disabled:not(:checked)+label:before,  [type="radio"]:disabled:checked+label:before {  box-shadow: none;  border-color: #8c8c8c;  background-color: #878787;  }  [type="checkbox"]:disabled:checked+label:after,  [type="radio"]:disabled:checked+label:after {  color: #555555;  }  [type="checkbox"]:disabled+label,  [type="radio"]:disabled+label {  color: #8c8c8c;  }  /\* accessibility \*/  [type="checkbox"]:checked:focus+label:before,  [type="checkbox"]:not(:checked):focus+label:before,  [type="checkbox"]:checked:focus+label:before,  [type="checkbox"]:not(:checked):focus+label:before {  border: 1px dotted #f6f6f6;  }  /\* hover style just for information \*/  label:hover:before {  border: 1px solid #f6f6f6 !important;  }  /\*=== Customization ===\*/  /\* radio aspect \*/  [type="checkbox"]:not(:checked)+label:before,  [type="checkbox"]:checked+label:before {  border-radius: 3px;  }  [type="radio"]:not(:checked)+label:before,  [type="radio"]:checked+label:before {  border-radius: 35px;  }  /\* selected mark aspect \*/  [type="checkbox"]:not(:checked)+label:after,  [type="checkbox"]:checked+label:after {  content: '✔';  top: 0;  left: 2px;  font-size: 14px;  }  [type="radio"]:not(:checked)+label:after,  [type="radio"]:checked+label:after {  content: '\2022';  top: 0;  left: 3px;  font-size: 30px;  line-height: 25px;  }  /\*=== 9. Misc ===\*/  .logo {  padding: 15px 0;  font-size: 25px;  color: #aaaaaa;  font-weight: bold;  }  #btnRegister {  float: right;  } |

### style.css

|  |
| --- |
| styles.css |
| 1. Create **styles.css** in the static/css folder and copy the following codes:  .row.content {  height: 550px  }  .sidenav {  background-color: #212020;  color: #c4c2c2;  height: 800px;  }  a {  color: white;  }  #navigations {  font-size: 1.5rem;  color: white;  }  .panel>.panel-heading {  background-image: none;  background-color: #3b3b3b;  color: white;  }  .active {  background-color: #3b3b3b;  border-radius: 10px;  } |

|  |
| --- |
| smartdoorbellWeb |
| .  |  ├── certs  | ├── certificate.pem.crt  | ├── private.pem.key  | ├── public.pem.key  | └── rootca.pem  |  ├── static  | └── css  | ├── login.css  | ├── register.css  | ├── styles.css  | └── welcome.css  |  ├── templates  | ├── dashboard.html  | ├── lcd.html  | ├── light.html  | ├── login.html  | ├── picture.html  | ├── register.html  | └── welcome.html  |  └── server.py |

# Section 9

# Running

Web Application

|  |
| --- |
| Task |
| 1. In the project directory for the web application, run  |  | | --- | | sudo python server.py | |

Smart Doorbell

|  |
| --- |
| Task |
| 1. In the project directory for the raspberrypi, run  |  | | --- | | sudo python ane\_doorbell.py | |

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