Final Project: *Eye Blinking Detection and Remote Access to IoT Sensors and Devices User Manual*

Team Hard Coders
Raymond Fey, Javier Garcia, Ifeanyi Anyika
1612920, 1266111, 2051191

Description

This project will allow a single client to use the eye-blinking detection application on the Raspberry Pi to control certain aspects of Internet of Things (IoT) devices connected to them, along with requesting help from a remote user. The remote user will be able to view certain sensor data, send messages, and remotely control the IoT devices from a web application on an Android smartphone.

The motivation for this project stems from the need to establish efficient communication between medical personnel and patients that have trouble speaking or moving their hands to use communication devices, and can also be extended to the deaf and people with other mental disorders. Being able to bridge the gap in fluent communication especially with technology and IoT will help medical personnel assist their patients more efficiently by enabling prompt responses.

Guide

Setting up for the first time

It will be necessary to have the following devices or components ready to run this application for the first time.

- Raspberry Pi 4 (Any RAM size)
- USB Webcam or any Camera using the Native Raspberry Pi Camera Interface
- BME280 Environmental Sensor
- 3x LEDs
- 3x 1000 Ohm Resistors
- Jumper wires
- Breadboard

A basic introduction on how to set up the Raspberry Pi for initial use is linked below.

https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up

After installing Raspberry Pi Desktop OS onto the Raspberry Pi and fully setting up the device, it is required to install some software updates and packages and connect several components to the Raspberry Pi before usage of the program.

Step 1

- The first step is to update the software dependencies on the Raspberry Pi

- Click on the Terminal icon on the top left part of the display as seen in **Figure 1**. This should be the fourth icon from the left and should look like a black box with an arrow

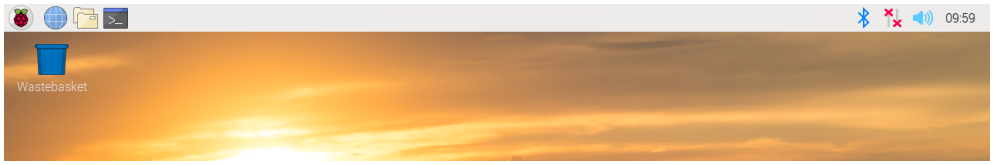

Figure 1. Raspberry Pi Desktop OS Default Interface

- Type and run the following command in the Terminal window as seen in **Figure 2**: `sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade`

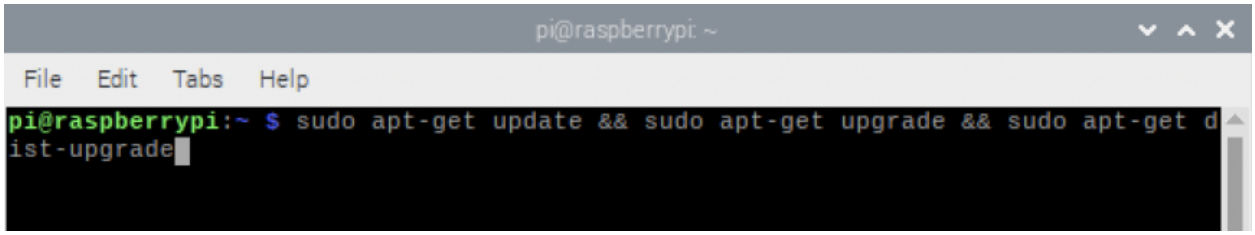

Figure 2. Terminal window to update software dependencies and other miscellaneous files

- You may have to type `Yes` when prompted
- You may have to reboot the system if prompted

Step 2

- Once Step 1 is complete, several software libraries and dependencies are required to be installed. Please try to follow the commands in order.
- In the Terminal, please run: `sudo apt install python3 idle3 && python3-pip`
- Please confirm you are running Python version 3.7.3 or above. To check this, please run the following command: `python3 --version`
- Next install OpenCV by running: `pip3 install opencv-contrib-python`
- Then PyQt5 can be downloaded and installed. This process may take several hours. To install, begin with running `sudo apt-get install qt5-qmake` and following by running `pip3 install PyQt5`
- The last small software libraries can then be installed by running
    - `pip3 install Flask`
    - `pip3 install netifaces`
    - `pip3 install adafruit-circuitpython-bme`

    - `pip3 install datetime`
    - `pip3 install numpy`
    - `pip3 install sockets`
    - `pip3 install schedule`
    - `pip3 install multiprocessing`
    - `pip3 install pynput`
- It may be required to reboot the system

Step 3

- Several interfaces on the Raspberry Pi need to be enabled.
- Click on the Raspberry Logo on the top-left on the display and navigate to Preferences then Raspberry Pi Configuration as seen in **Figure 3**.

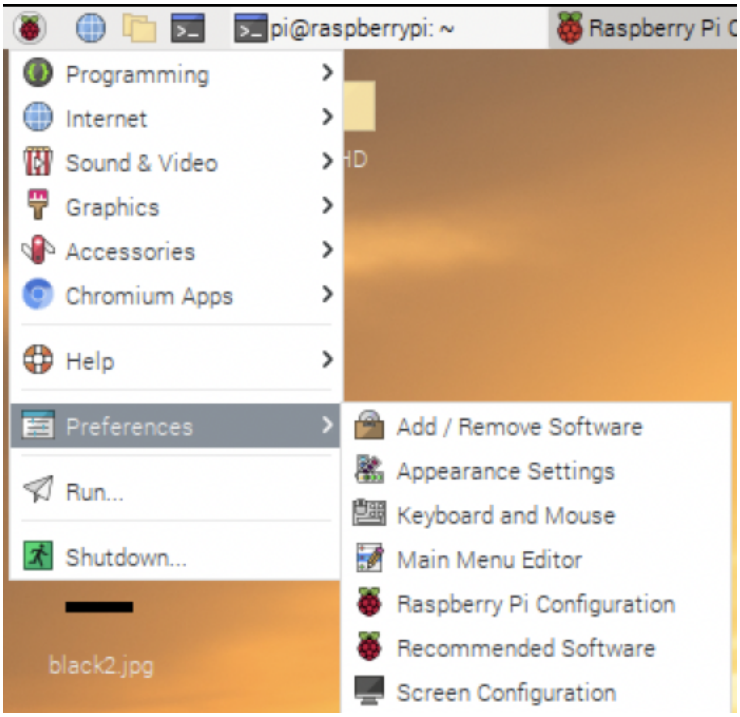

Figure 3. Finding the Raspberry Pi Configuration Panel

- Under the Raspberry Pi Configuration window, please navigate to Interfaces and enable Camera and I2C and click OK as seen in **Figure 4**. You may need to reboot the Raspberry Pi for settings to take effect.

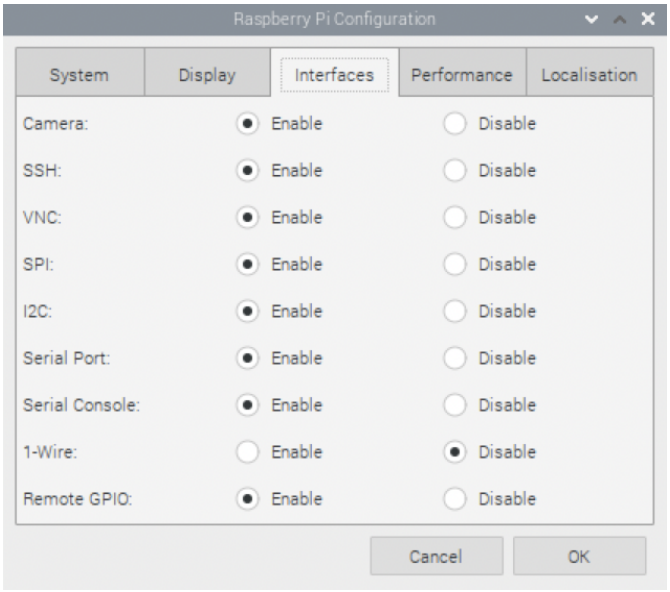

Figure 4. Enabling I2C and Camera under Raspberry Pi Configuration Panel

Step 4

- Please have a breadboard ready to connect wires and components.
- On the BME280 Environmental Sensor, connect the following:
    - VCC to Pin 2 (or any 5V power on the Raspberry Pi) on the General Purpose IO
    - GND to Pin 6 (or any Ground on the Raspberry Pi) on the GPIO
    - SCL to Pin 5 on the GPIO
    - SDA to Pin 3 on the GPIO
    - SDO to Pin 1 (or any 3V3 power on the Raspberry Pi) on the GPIO
        - It is very important to have SDO connected to a 3.3V source for the addressing mode to be set correctly.
- For the LEDs connect Pins 35, 37, and 39 from the Raspberry Pi to a 1000 ohm resistor in series with each LED. At the end of each LED, connect to any GROUND pin on the Raspberry Pi's GPIO.
- Please refer to **Figure 5** for the Raspberry Pi 4's GPIO pin layout.
- Please note that VCC or GND pins are not fixed; however, for all other pins, please adhere to the instructions.

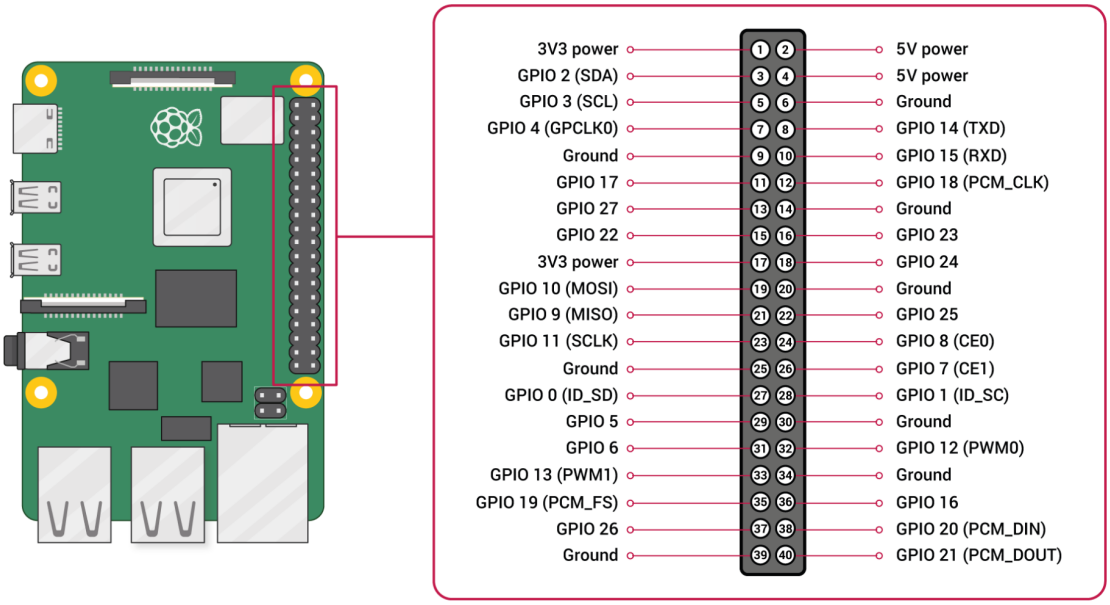

Figure 5. Raspberry Pi 4 General Purpose IO Pin Layout

Step 5

On any Windows/macOS/Linux system, please install Android Studio onto your system. Please go to this link to download Android Studio… https://developer.android.com/studio

Step 6

Download the Github Repository to a local directory on your system.

Small Configuration Revisions

Step 1

Open up a Terminal window on the Raspberry Pi and change to the `AHD-Final-Project` then `src` then `Raspberry Pi App` directory.

Step 2

In the Terminal, run the command: `python3 GUI.py`. Pay attention to the output of the Terminal and you should see an IP and the port number of your webserver. The output should print out something similar to `Running on http://192.168.X.X:8080/ (Press CTRL+C to quit)`

In **Figure 6**, the IP address of the Raspberry Pi's Flask web server is 192.168.1.13.

Please copy the IP down somewhere convenient. The IP is the numbers between the two colons ":".

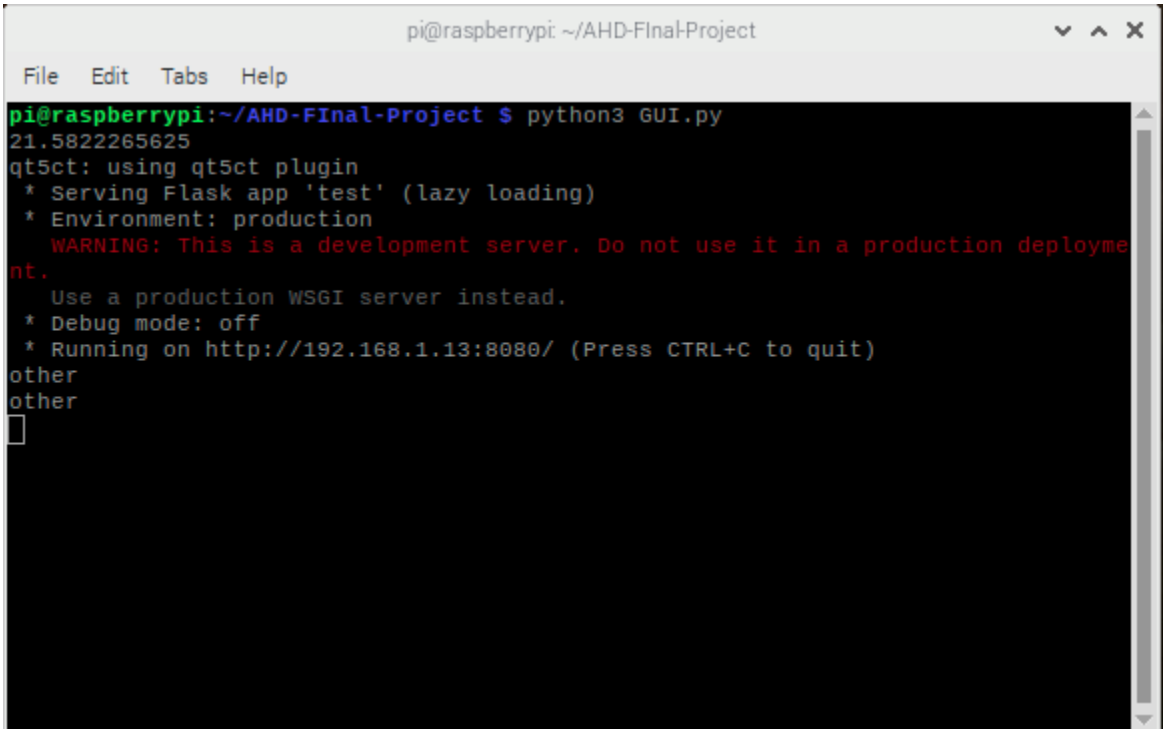

Figure 6. Running GUI.py to obtain the IP address of the Flask webserver

Step 3

In Android Studio, please navigate to the directory, `/AHD-Final-Project/Android Web App/My Application/app/src/main/java.com/example/myapplication` and find the file named `MainActivity.kt`.

In this file, on line 18 in **Figure 7**, replace the IP address `https://192.168.142.30:8080` to the IP address observed in Step 2 and save the file.

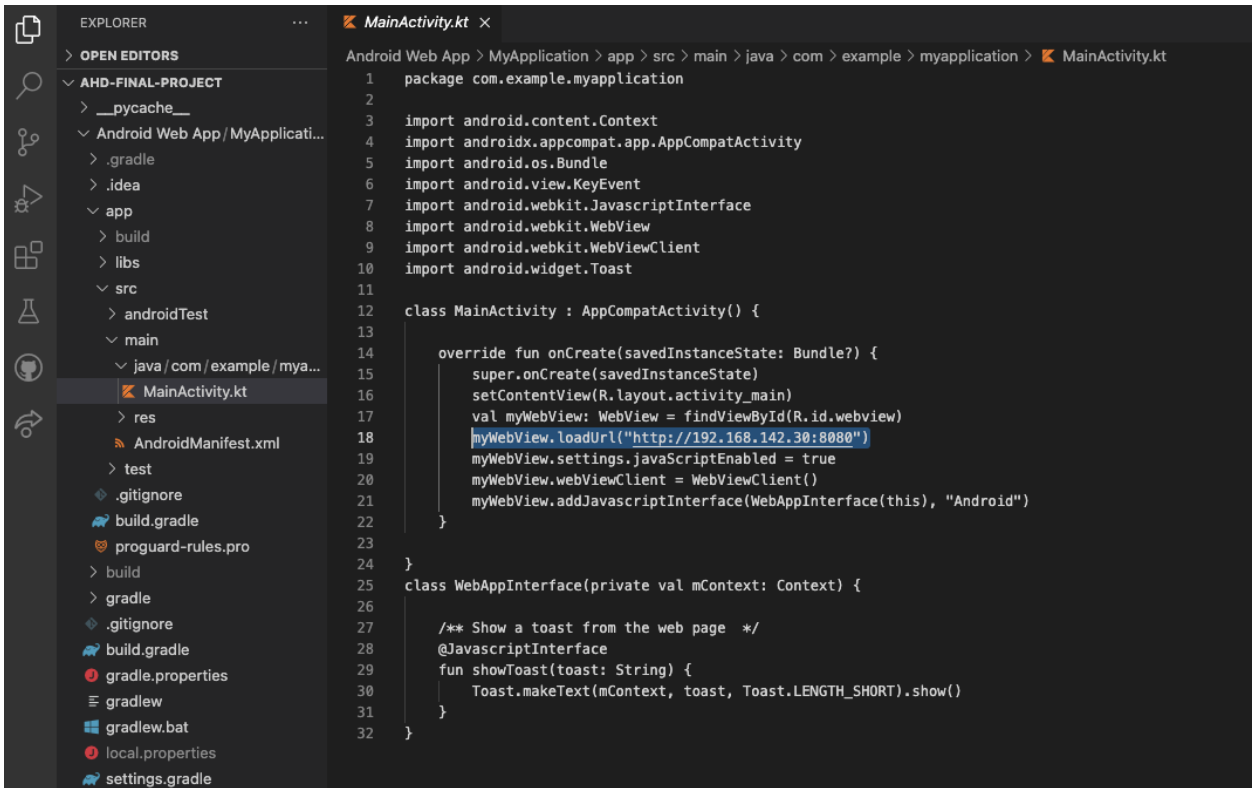

Figure 7. Changing the IP address in MainActivity.kt

Running the Application

To start the application, run the command: `python3 GUI.py` in the Terminal as seen in **Figure 6**. After a few moments, a window will show up similar to **Figure 8**.

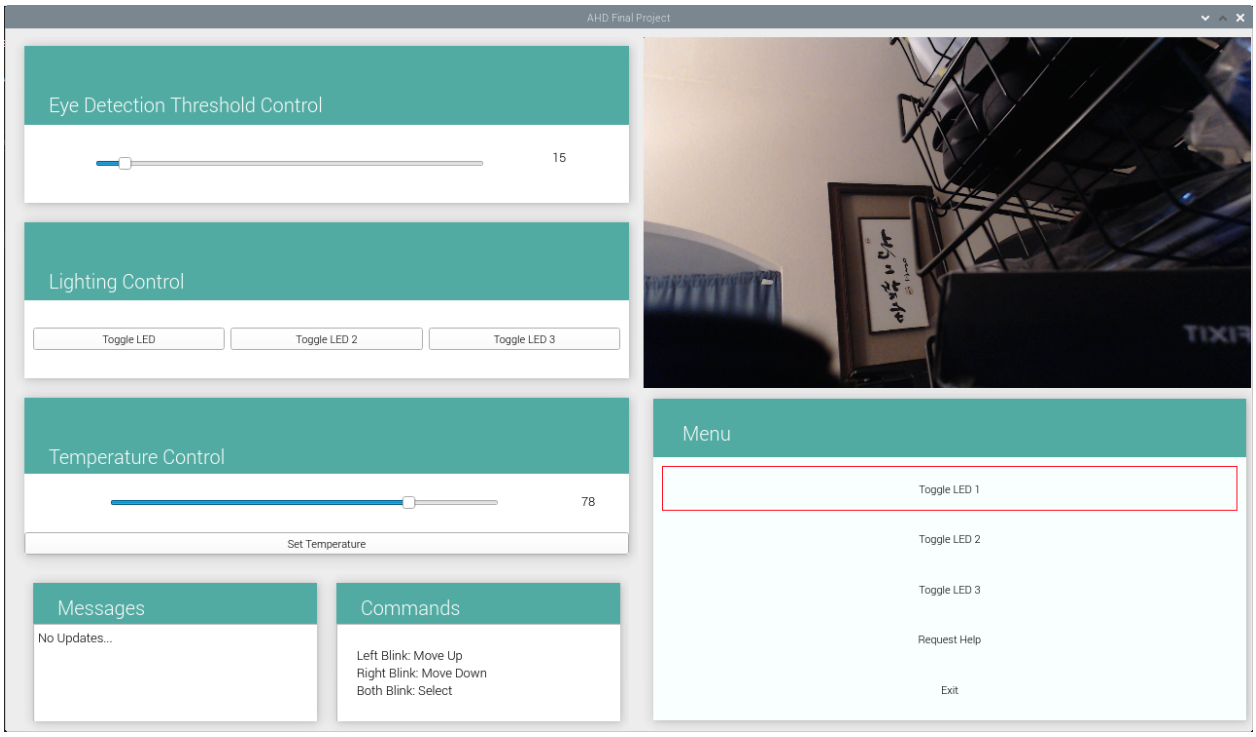

Figure 8. Raspberry Pi Local Application using PyQt5

Using the Application

Using the application is simple, as there are only a few features that were implemented. The application is split into two parts, manual control and an eye blinking control element.

- Manual Controls
    - For the eye detection software, the threshold value can be changed to help the software algorithm to detect your eyes better. To change this value, use the slider under the card, Eye Detection Threshold Control.
    - The lights in the room can be toggled by pressing the Toggle LED buttons under the Lighting Control card.
    - The temperature in the room can be set by dragging the slider to the desired temperature and then pressing the Set Temperature push button. These changes will also immediately reflect on the webserver; however, the slider and the label will update every 5 seconds if the remote user decides to change the temperature through the webserver.
- Eye Blinking Controls

- On the right bottom section of the application, there is a Menu and a red box to show the current selection.
- To move up the red box, blink your LEFT eye for about a second.
- To move down the red box, blink your RIGHT eye for about a second.
- To select and perform the current option highlighted by the red box, blink BOTH of your eyes for about a second
- **Example**: If you want to perform the Request Help option, you should blink your right eye three times then proceed to blink both eyes.
- **NOTE**: The red box will always start at Toggle LED 1 after starting the programming and will remain at whatever option you select. Also, a small guide on how to use the eye blinking commands are in the card named Commands

Using the Web Server

The web server should automatically start when running `python3 GUI.py`. In your browser's URL address bar, enter the IP address and port number of the webserver. You should see something similar to **Figure 9**.

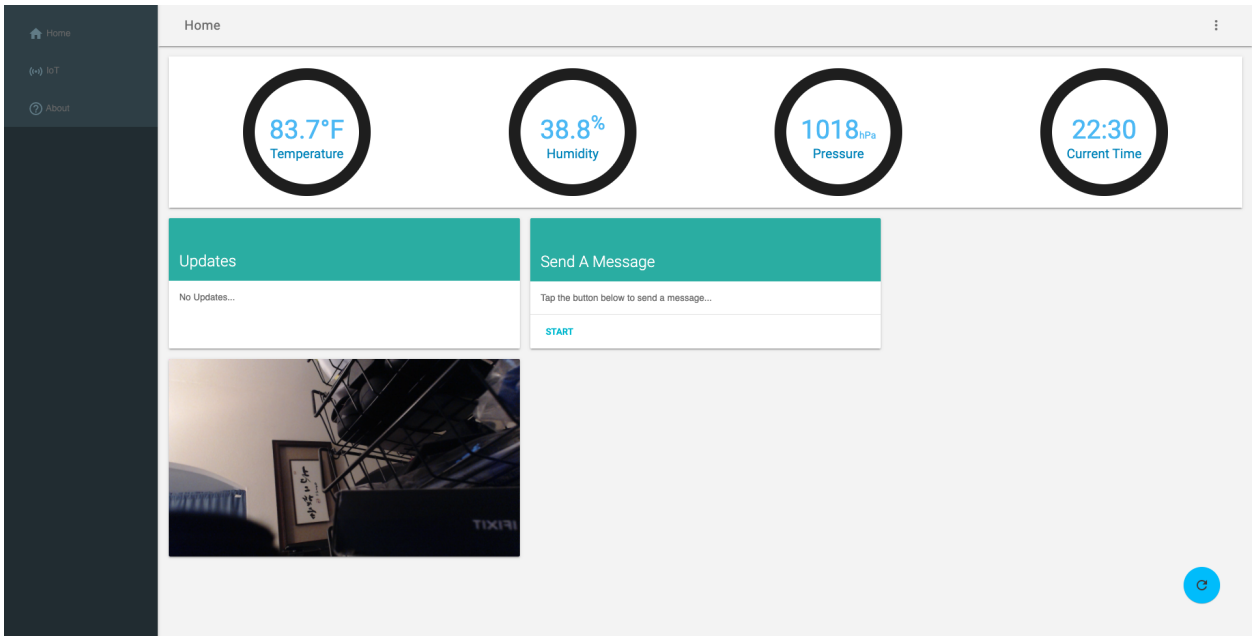

Figure 9. The home page of the webserver

On the homepage of the webserver, you can see useful information such as the current temperature, humidity, pressure, and time of the client's location. There are also options to see messages sent by the client from the Raspberry Pi and the ability to send a message back to the client.

It is unfortunate that automatic updates to the site are not possible in this code implementation; therefore, periodically refreshing the site manually is required by the remote user. To refresh, click on the blue circle button on the bottom right of the page. This will GET and update the page for you. This will also grab a new frame of the video capture from the client's side.

To send a message, press the START button under the Send A Message Card. A popup should appear as seen in **Figure 10** and a message can be written and sent by pressing the SEND button.

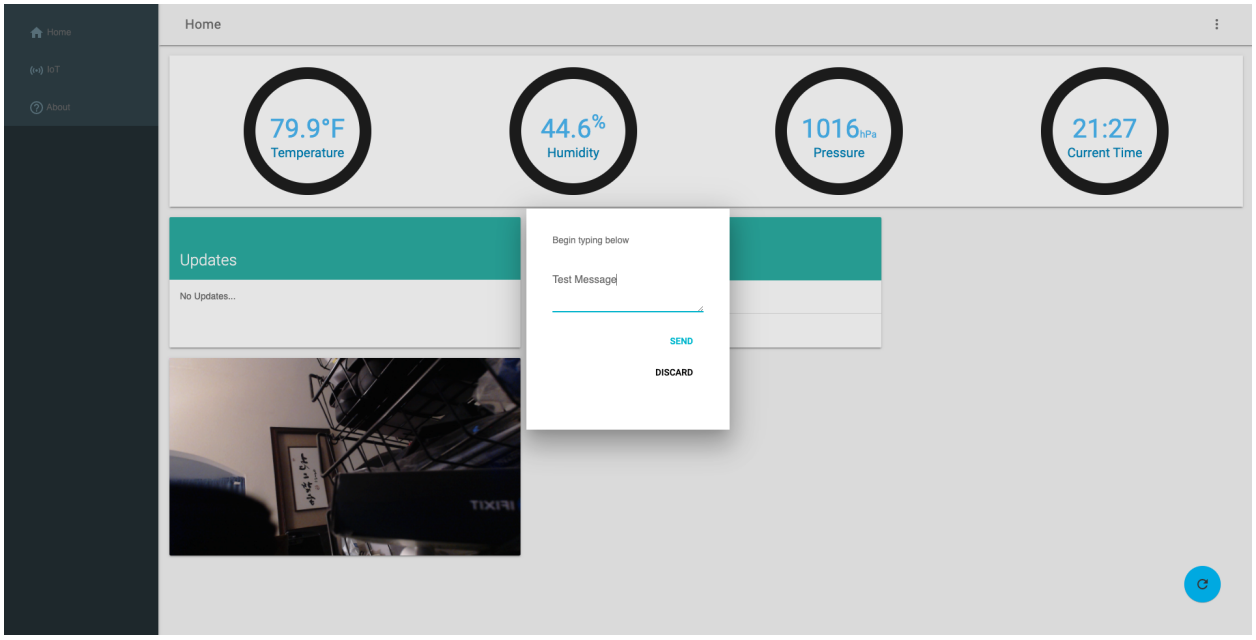

Figure 10. Sending a message through the webpage

After pressing the send button, the message should automatically update at the client's local Raspberry Pi application as seen in **Figure 11**.

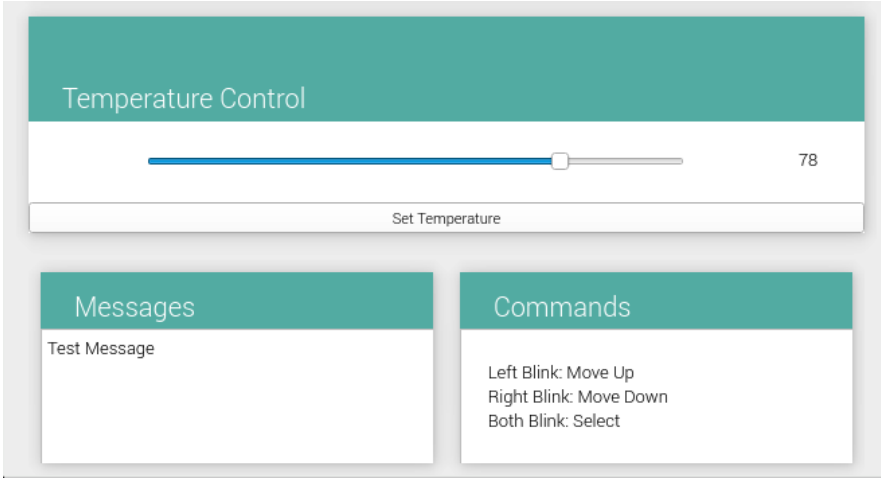

Figure 11. Received message on the local Raspberry Pi Application

To remotely control the devices such as the LEDs or the temperature, you can do so from the website. Go to the sidebar and click on IoT as seen in **Figure 12**. Conveniently, different cards allow you to toggle three different LEDs and set the temperature. To toggle an LED, select TOGGLE LED for any card and the status of the LED will be updated to show whether it is on or off. To change the temperature of the client's room, change the slider to the desired temperature and press the set temperature button.

Please note that if the client decides to toggle the lights or change the temperature, you will need to refresh as it does not update automatically. Similar to the home page, use the blue refresh button on the bottom right of the screen to refresh the changes, as using the browser's refresh button may duplicate responses and may cause unwanted changes or damages to the GPIO pins.

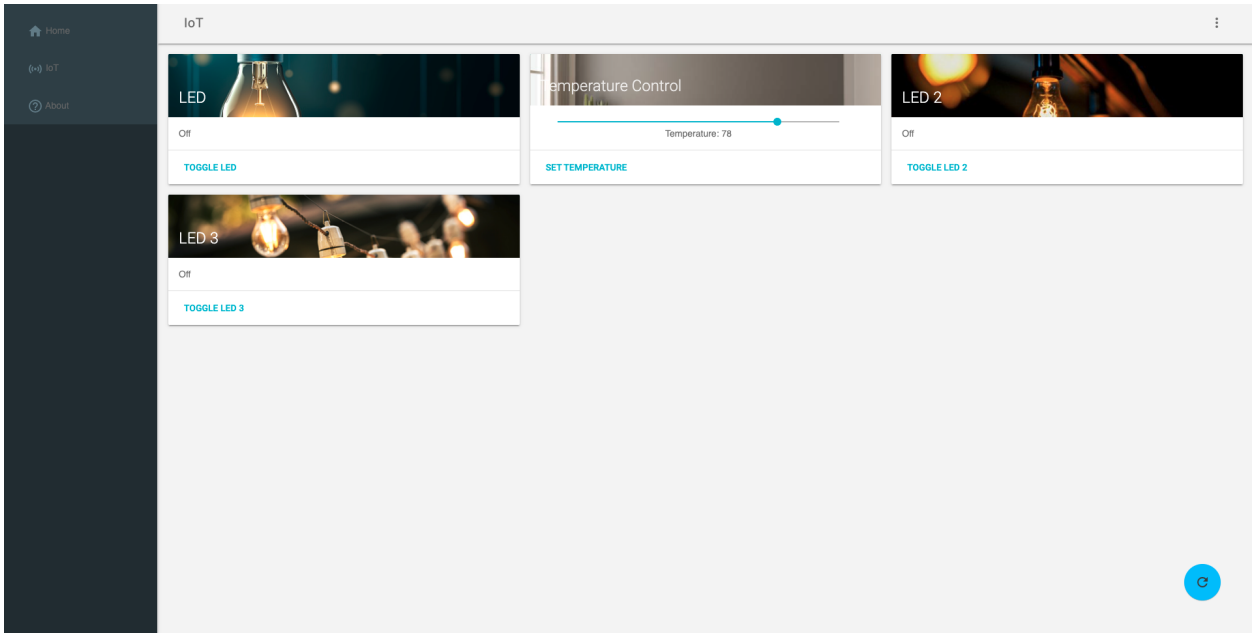

Figure 12. Controlling devices at the client's side through the web page

Using the Android Application Under Android Studio

Before running the Android Application, make sure to start the Python script from the **Running the Application** section. To run the Android Application, press the green triangle, or the "play" button, as seen in **Figure 13**, to allow Android Studio to build and compile.

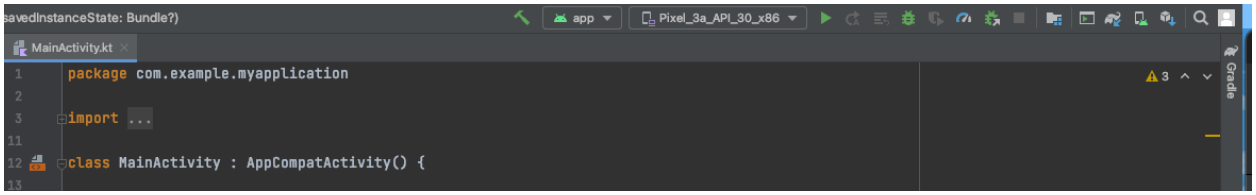

Figure 13. Running the Android Application in Android Studio

An emulator that looks like a Pixel 3A smartphone should show up as seen in **Figure 14**. Once fully loaded, you can use the app similar to the website, as this app is purely a web app.

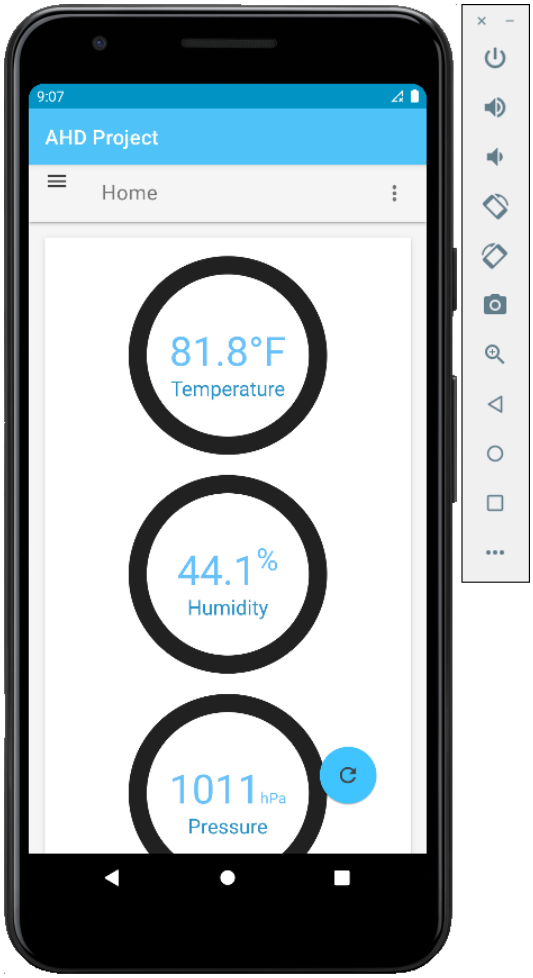

Figure 14. Android App initial state

Quitting the Application

- There are two methods of fully quitting the application at once
    - Method 1: Eye blinking method.
        - By continuously blinking your right eye, the red box should go down to the Exit option on the Menu within the Raspberry Pi application.
        - Blink both eyes and the application should close.
    - Method 2: Force closing by pressing CTRL+C in the terminal window