

# ชื่อโครงการ JTM for you health

## สมาชิกในกลุ่ม

- 64106107 อัมมกิตติ์ โชติช่วง
- 64107899 ปณัฒน์ เอี่ยมน้ำ
- 64113889 ศุภณัฐ คุ่มรักษ์

## หน้า Website การแสดงผล



## บทนำ

ทางกลุ่มผู้จัดทำโครงการได้มองเห็นถึงความสำคัญเรื่องสภาพแวดล้อมและความพึงพอใจของผู้คน ซึ่งเป็นปัจจัยสำคัญที่ส่งผลต่อคุณภาพชีวิตและประสิทธิภาพการทำงาน โดยความเหมาะสมของสภาพแวดล้อมนั้นประกอบไปด้วย อุณหภูมิ ความชื้นภายในอาคาร และที่สำคัญคือ ค่าฝุ่น PM2.5 และ PM10 ที่ส่งผลต่อสภาวะทางเดินหายใจ ซึ่งทางกลุ่มจึงจัดทำอุปกรณ์ที่ตรวจจับค่า อุณหภูมิ ความชื้น และค่าฝุ่น ที่เหมาะสม เพื่อเป็นแนวทางในการป้องกันหากเกิดเหตุการณ์ที่ค่าดังกล่าวผิดปกติ เพื่อเพิ่มความเป็นอยู่ให้ดีขึ้นกับผู้ใช้งานที่กำลังดำเนินกิจกรรมในขณะนั้น

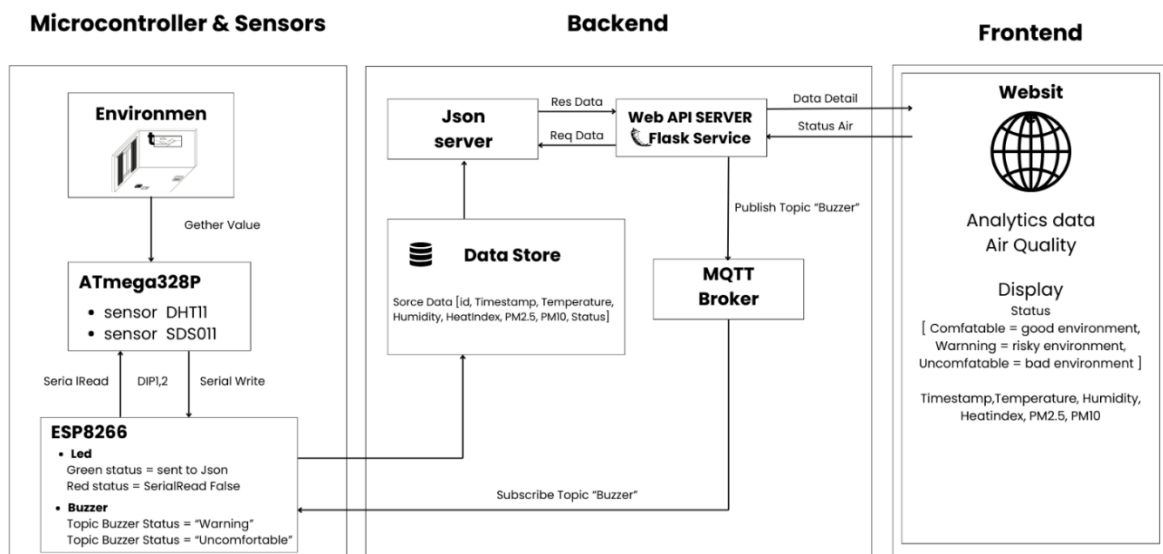
เนื่องด้วยปัจจุบัน สภาพแวดล้อมหรือสภาพอากาศนั้นมีความแปรปรวนเป็นอย่างมากซึ่งส่งผลเสียให้คนที่เป็นโรคมะเร็งหรือคนปกติเกิดสภาวะติดขัดทางเดินหายใจ หรือหายใจไม่สะดวก และรุนแรงถึงขั้นหายใจไม่ได้ และเพียงแค่ค่าอุณหภูมิที่แสดงอยู่บนแอปพลิเคชันในปัจจุบันนั้นก็เพียงพอที่จะบอกได้ว่าสภาพแวดล้อมที่ผู้ใช้งานอยู่นั้นเหมาะสมหรือไม่เหมาะสม ซึ่งจากปัญหาเหล่านี้ทางกลุ่มผู้จัดทำจึงมุ่งเน้นในการผลิตอุปกรณ์ IoT ที่สามารถตรวจจับค่าอุณหภูมิ ความชื้น และค่าฝุ่น โดยนำค่าอุณหภูมิและความชื้นมาคำนวณค่าดัชนีความร้อน เพื่อเป็นตัวช่วยในการรับมือกับสถานการณ์ที่จะเกิดขึ้นกับอุณหภูมิที่ผู้ใช้รับรู้ได้จริง ณ ขณะนั้น

กลุ่มเราจึงมุ่งเน้นในการทำอุปกรณ์ IoT เพื่อตรวจจับค่าจากสิ่งแวดล้อมได้แก่ ค่าอุณหภูมิ ความชื้น ค่าฝุ่น PM2.5 และ PM10 โดยจะแสดงผลขึ้นเว็บไซต์เพื่อบอกคุณภาพสภาพอากาศ ณ ขณะนั้นว่ามีผลเป็น ปกติ แย่ และแย่มาก ซึ่งจะมีการแจ้งเตือนด้วยเสียงผ่านอุปกรณ์ IoT เพื่อจะให้ผู้ใช้งานรับรู้ถึงสภาพอากาศที่ไม่เหมาะสม ณ ขณะนั้น ซึ่งสามารถช่วยการตัดสินใจในการแก้ปัญหาเรื่องสภาพแวดล้อมด้วยตัวเอง

สิ่งที่เรามุ่งเน้นคือการนำเทคโนโลยี IoT ที่มีอยู่มาใช้ในการตรวจจับค่าอุณหภูมิ ความชื้น มาคำนวณค่าดัชนีความร้อน และค่าฝุ่นที่มีอยู่ในสภาพแวดล้อมนั้นมาประมวลผล และแสดงผลค่าความเหมาะสมของสภาพภูมิอากาศว่าอยู่ในเกณฑ์อะไร เพื่อให้ผู้ใช้งานสามารถรับมือกับสถานการณ์ที่สภาพแวดล้อมนั้นเริ่มเปลี่ยนแปลง และส่งผลเสียต่อตัวผู้ใช้งานเอง ดังนั้นโครงงานของเราจึงเป็นตัวช่วยในการตัดสินใจในการรับมือกับสถานการณ์ที่สภาพแวดล้อมนั้นมีแนวโน้มที่ส่งผลเสียให้กับผู้ใช้งาน

## การออกแบบระบบ

### สถาปัตยกรรมระบบ



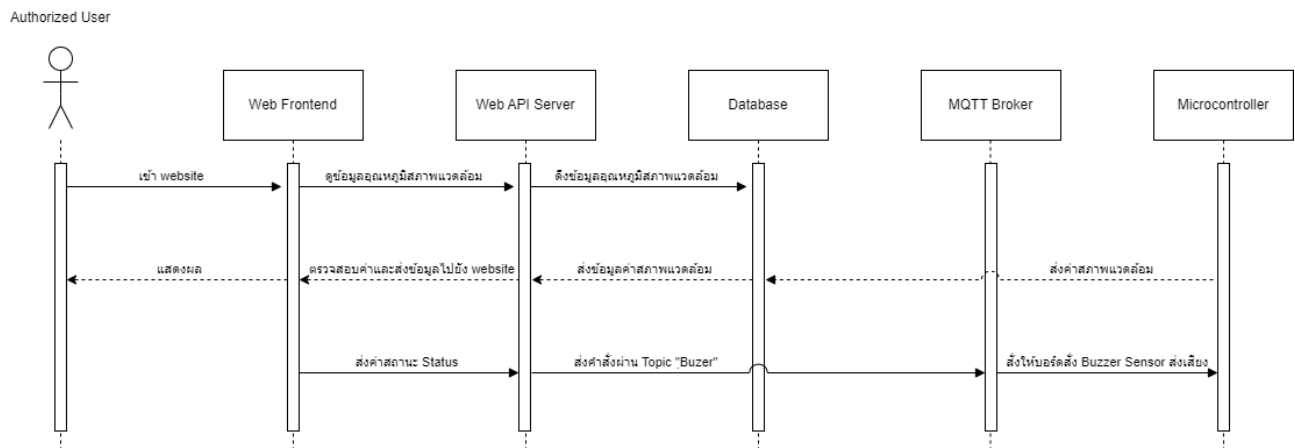
จากสถาปัตยกรรมระบบนี้ เป็นสถาปัตยกรรมของระบบ JTM ที่แบ่งเป็น 3 โมดูลดังนี้

โมดูลที่ 1 คือ Microcontroller & Sensors เป็นโมดูลที่เกี่ยวกับอุปกรณ์ด้าน Hardware ต่าง ๆ โดยส่วนของ ATmega32P นั้นจะทำหน้าที่ในการเก็บค่าข้อมูลที่ได้จาก Sensor DH11, Sensor SDS011 และทำการเช็คค่า Status หากมีค่าดัชนีความร้อนหรือ PM2.5 ที่อยู่ในเกณฑ์เฝ้าระวังหรืออันตราย จากนั้นจะส่งค่าไปยัง ESP8266 ที่จะแสดงสถานะส่งข้อมูลผ่าน LED ว่าสามารถอ่านข้อมูลจาก ATmega32P และส่งข้อมูลเข้า json server ได้หรือไม่

โมดูลที่ 2 คือ Backend เป็นโมดูลที่เกี่ยวกับการประสานงานการรับส่งข้อความระหว่างไคลเอนต์ต่างๆ โดยประกอบด้วยส่วนของ Json server ทำหน้าที่ในการรับและกรองข้อความจาก Microcontroller เพื่อใช้ในการติดต่อระหว่างโมดูลและจัดเก็บข้อมูล จากนั้น Flask server จะทำหน้าที่ในการเรียกใช้ API จาก Json Server เพื่อทำหน้าที่เป็นตัวติดต่อระหว่าง Json server และ Frontend

โมดูลที่ 3 คือ Frontend เป็นโมดูลที่เกี่ยวกับการแสดงผลข้อมูล สถานะต่าง ๆ ของค่าที่ได้รับมาจาก Sensor จะทำหน้าที่เป็นส่วนของการแสดงผลซึ่งในส่วนนี้จะเป็นการแสดงผลข้อมูลที่ได้รับมาในส่วนของ Json Server ประกอบไปด้วย ค่าอุณหภูมิ ค่าความชื้น ค่าดัชนีความร้อน ค่าฝุ่น PM2.5 PM10 Timestamp และคุณภาพอากาศแบบ Realtime ทุกๆ 5 วินาที นอกจากนี้จะมีการส่งค่าคุณภาพอากาศกลับไปยังบอร์ด เพื่อแจ้งเตือนคุณภาพอากาศ Warning และ Uncomfortable

### สถาปัตยกรรมซอฟต์แวร์



Module สำหรับการใช้งานผ่าน Web Frontend โดยผู้ใช้งานจะต้องทำงานเข้าใช้งาน Website ก่อน จากนั้นระบบจะทำการดึงค่าจาก Web API Server มาแสดงผลบนหน้า Website โดยจะมีการอัปเดตค่าสถานะและสภาพแวดล้อมทุกๆ 5 วินาที

ระบบจะส่งค่า Status จาก Website ไปประตู่ผ่าน MQTT Broker ไปยัง Microcontroller ซึ่งจะสั่งให้ Buzzer Sensor ทำงานเพื่อส่งเสียงบ่งบอกถึงระดับความอันตรายในสภาพแวดล้อมนั้น โดย Module นี้ประกอบด้วย 5 องค์ประกอบหลัก ได้แก่

1. Authorized User: ผู้ใช้งานระบบ
2. Web Frontend : หน้าเว็บไซต์สำหรับแสดงค่าสถานะและสภาพแวดล้อม
3. Web API Server : บริการ API สำหรับ Web Frontend ในการรับ-ส่งข้อมูล

4. Database : ฐานข้อมูลสำหรับเก็บข้อมูลสภาพแวดล้อม
5. MQTT Broker : ตัวกลางสำหรับรับส่งข้อมูลสถานะระหว่าง Web API Server และ Microcontroller

### โครงสร้างข้อมูล

ข้อมูลที่ได้รับจากเซนเซอร์ต่าง ๆ จะถูกจัดเก็บในรูปแบบโครงสร้างของ Json ที่จะประกอบไปด้วยข้อมูลของ Temperature, Humidity, HeatIndex, PM2.5, PM10, Status, timestamp และ id ตามลำดับ ที่ถูกจัดเก็บใน Module ชื่อ ESP8266

```
{
  "ESP8266": [
    {
      "Temp": 23.29999924,
      "Humid": 95,
      "HeatIndex": 24.17000008,
      "PM25": 5.400000095,
      "PM10": 8.699999809,
      "Status": "Comfortable",
      "timestamp": "23-12-2023 23:49:19",
      "id": 1
    }
  ]
}
```

### Data Dictionary

Attribute Name	Description	Data Type	Example
id	id ของข้อมูลที่น่าเข้า	int	1
Timestamp	เวลาที่นำเข้าข้อมูล	timestamp	12-12-2023 20:32:11

Temp	ค่าของอุณหภูมิ	float	24.02
Humid	ค่าของความชื้น	float	62.00
HeatIndex	ค่าของอุณหภูมิจากมนุษย์	float	24.06
PM2.5	ค่าของฝุ่นขนาดไม่เกิน 2.5 ไมครอน	float	12.80000019
PM10	ค่าของฝุ่นขนาดไม่เกิน 10 ไมครอน	float	14.39999962
Status	ค่าที่บ่งบอกถึงคุณภาพอากาศ ณ ปัจจุบัน	String	Comfortable

## การพัฒนาระบบ

### Board

1. ATmega328P ทำหน้าที่ในการวัดค่าจากเซนเซอร์ต่าง ๆ
2. ESP8266 ทำหน้าที่ในการรับค่าจาก ATmega328P และส่งขึ้นไปยัง Json Server

### Sensors

1. DHT11 ทำหน้าที่ในการรับค่า Temperature, Humidity และคำนวณค่าดัชนีความร้อน จากสิ่งแวดล้อม
2. SDS011 ทำหน้าที่ในการรับค่า PM2.5, PM10 จากสิ่งแวดล้อม
3. Buzzer ทำหน้าที่ในการส่งเสียงแจ้งเตือนหากค่าสภาพแวดล้อมอยู่ในเกณฑ์อันตราย
4. Led ทำหน้าที่ในการแสดงสถานะไฟสีเขียวหากสามารถส่งค่าขึ้นไปยัง Json และสถานะไฟสีแดงหากไม่สามารถรับค่าจาก ATmega328P

### เครื่องมือ

1. ภาษา C ใช้ในการเขียนคำสั่งลงไปยัง Board เพื่อที่จะสามารถรับค่าและส่งค่าข้อมูล
2. JSON ใช้ในการเป็น server ในการรับค่าจาก Board
3. JavaScript ใช้ทำให้ website สามารถติดต่อกับผู้ใช้ได้
4. HTML ใช้ในการพัฒนาโครงสร้าง website เพื่อใช้ในการแสดงผล
5. CSS ใช้ในการตกแต่ง website

6. Python: Flask, requests, paho-mqtt, เป็น framework ในการเชื่อมต่อ IoT ขึ้นเว็บไซต์ สำหรับดึงข้อมูลจาก Json Server ร่วมกับส่งค่าสถานะไปยังตัว Board เพื่อแจ้งเตือนด้วยเสียงผ่านเซนเซอร์ Buzzer

## ไลบรารี

1. Bootstrap เฟรมเวิร์ค ที่ช่วยให้สามารถสร้างเว็บไซต์ใช้เพื่อสร้างองค์ประกอบการออกแบบต่างๆ
2. DHT.h: ใช้ในการอ่านข้อมูลเซนเซอร์วัดอุณหภูมิและความชื้นชนิด DHT และคำนวณค่าดัชนีความร้อน
3. SoftwareSerial.h: เปิดใช้งานการสื่อสารแบบอนุกรมบนขาอื่นๆ
4. SdsDustSensor.h: ใช้ในการอ่านข้อมูลเซนเซอร์วัดฝุ่นชนิด SDS
5. ESP8266WiFi.h: ใช้ในการเชื่อมต่อกับเครือข่าย Wi-Fi บนบอร์ด ESP8266
6. time.h: ใช้ในการทำงานเกี่ยวกับเวลาและวันที่
7. WiFiUdp.h: ใช้ในการสื่อสารแบบ UDP
8. NTPClient.h: ใช้ในการปรับเทียบเวลากับเซิร์ฟเวอร์ NTP
9. ESP8266HTTPClient.h: ใช้ในการส่งค่าขอ HTTP ผ่านบอร์ด ESP8266
10. ArduinoJson.h: ใช้ในการแยกวิเคราะห์และสร้างข้อมูลแบบ JSON

## การทดสอบ

โดยภาพนี้จะเป็นห้องที่ใช้ในการทดลอง

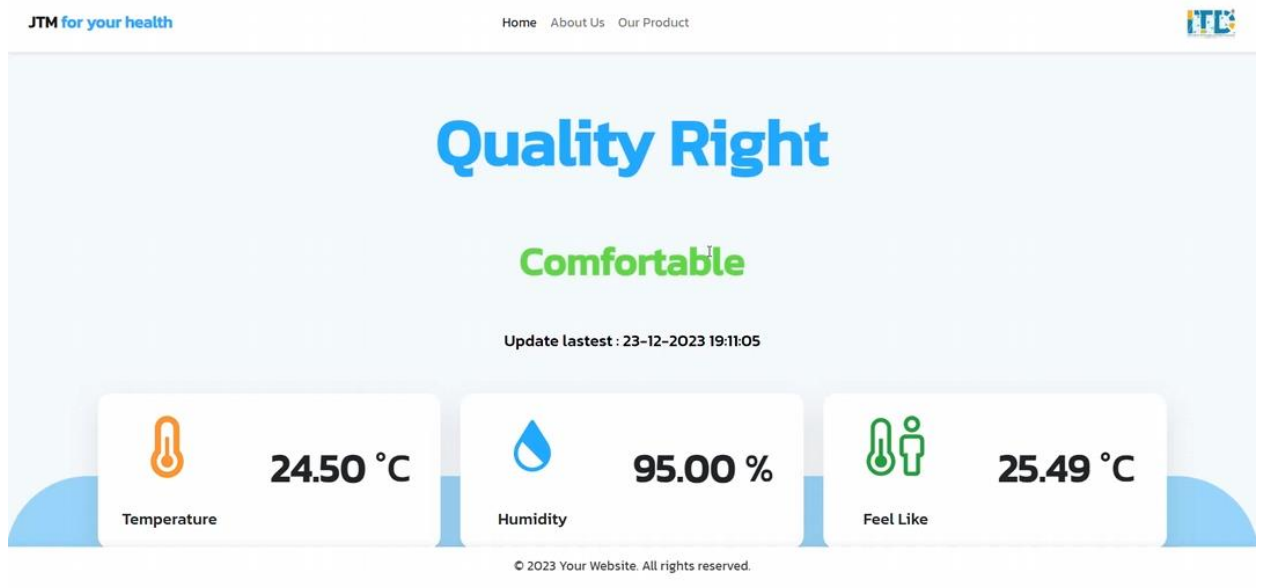


- Test Case #1

ทดสอบนำ JTM ไปไว้ในห้องที่ปิดหน้าต่างและมีการเปิดแอร์อยู่ที่ 25 องศา

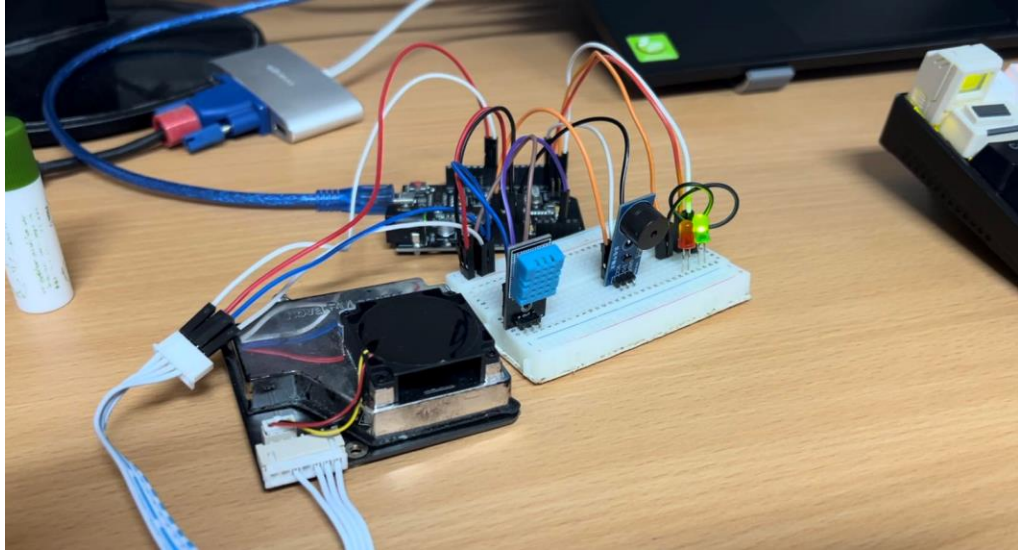


มีการเปิดแอร์และตั้งค่าแอร์อยู่ที่ 25 องศาและมีการปิดหน้าต่าง



ภาพผลลัพธ์การแสดงผล Status Comfortable





โดยในกรณีนี้จะมีการส่งแรงเตือนเป็นเสียงจาก sensor buzzer ที่จะมีเสียงเพื่อสื่อถึงการแจ้งเตือนควร  
เฝ้าระวังสภาพอากาศภายในห้อง

- Test Case #2

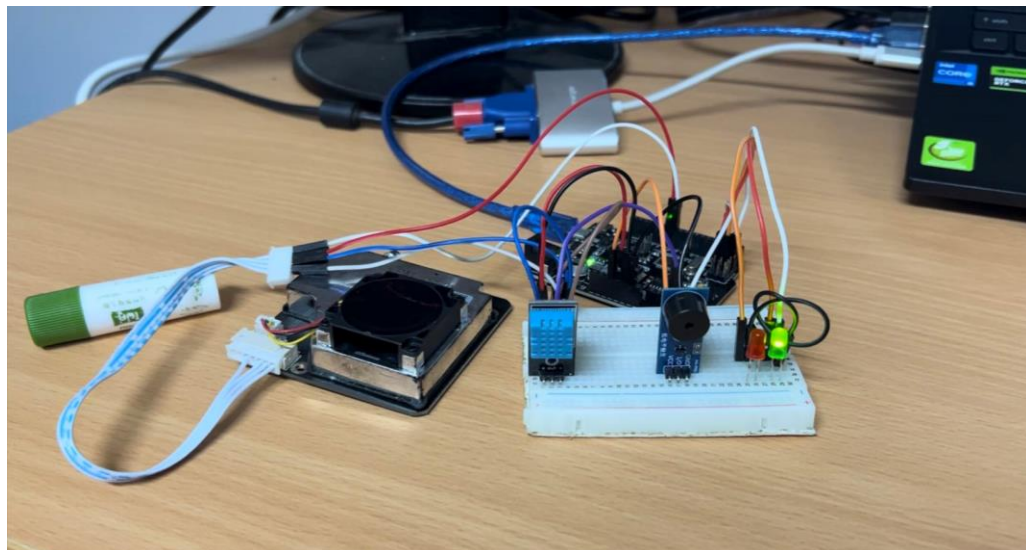
ทดสอบนำ JTM ไปไว้ในห้องที่มีการปิดแอร์และเปิดหน้าต่างเพื่อให้มีอากาศถ่ายเท



มีการปิดแอร์และมีการเปิดหน้าต่าง



ภาพผลลัพธ์การแสดงผล Status Warning



โดยในกรณีนี้จะมีการส่งแจ้งเตือนเป็นเสียงจาก sensor buzzer ที่จะมีเสียงเพื่อสื่อถึงการแจ้งเตือนควร  
เฝ้าระวังสภาพอากาศภายในห้อง

- Test Case #3

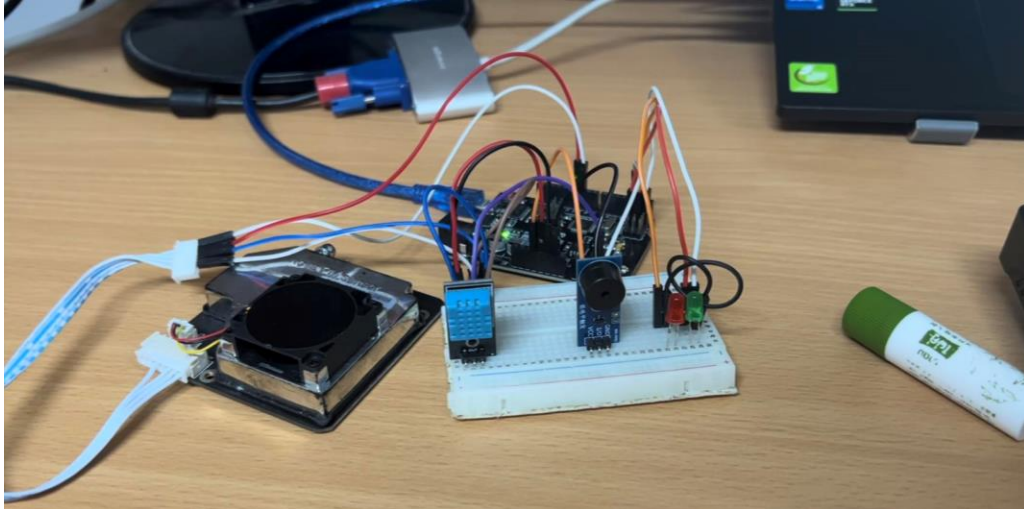
ทดสอบนำ JTM ไปไว้ในห้องที่มีการปิดแอร์และปิดหน้าต่างเพื่อไม่ให้มีอากาศถ่ายเท



มีการปิดแอร์และมีการปิดหน้าต่าง



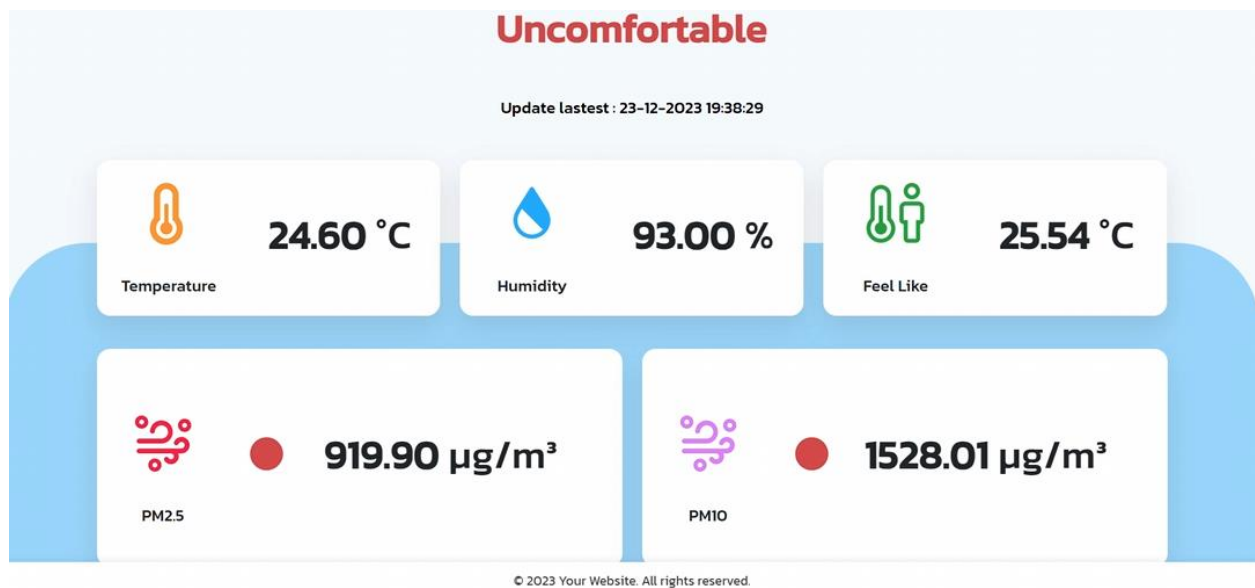
ภาพผลลัพธ์การแสดงผล Status Uncomfortable



โดยในกรณีนี้จะมีการส่งแจ้งเตือนเป็นเสียงจาก sensor buzzer ที่จะมีเสียงเพื่อสื่อถึงการแจ้งเตือนสภาพอากาศที่เป็นอันตรายภายในห้อง

- Test Case #4

ทดสอบนำ JTM ไปไว้ในห้องที่มีค่าของฝุ่น PM2.5 และ PM 10 ที่สูง



ภาพผลลัพธ์การแสดงผล Status Uncomfortable



โดยในกรณีนี้จะมีการส่งแจ้งเตือนเป็นเสียงจาก sensor buzzer ที่จะมีเสียงเพื่อสื่อถึงการแจ้งเตือนสภาพอากาศที่เป็นอันตรายภายในห้อง

### สรุปผลการทดสอบ

จากผลการทดลองทั้ง 4 กรณีที่เราได้มีการทดสอบไป พบว่าการทดสอบเป็นไปตามวัตถุประสงค์ที่เราได้ตั้งไว้บอร์ด ATmega32P สามารถรับค่าได้จาก Sensor และมีการส่งค่าไปยังบอร์ด ESP8266 เพื่อที่จะส่งค่าต่อไปยัง Json Server จากนั้น Flask Server ทำการดึง API ของ Json Server ต่อเพื่อนำไปแสดงผลบนเว็บไซต์แล้ว จากนั้นเว็บไซต์ก็สามารถแสดงค่าสถานะ Temp, Humid, HeatIndex, PM2.5, PM10, และสถานะ Status ได้ตามเกณฑ์ที่เราได้มีการกำหนดไว้จากแหล่งอ้างอิงในแต่ละกรณีตามที่เราได้ตั้งสมมุติฐานไว้