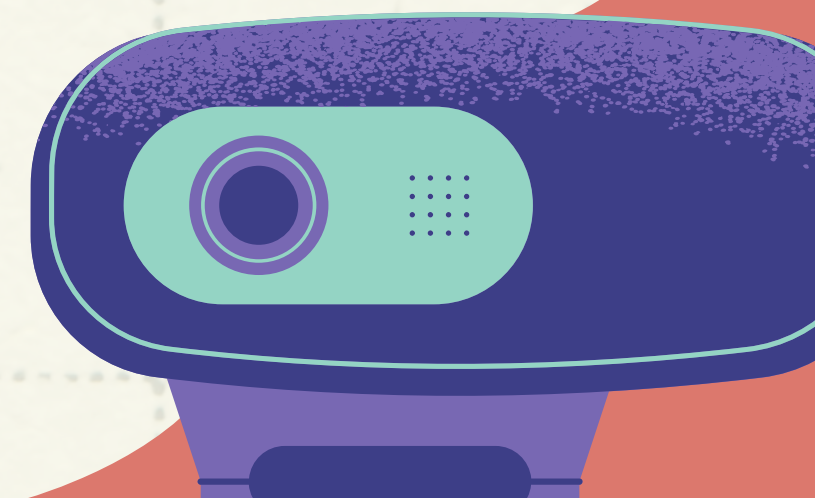
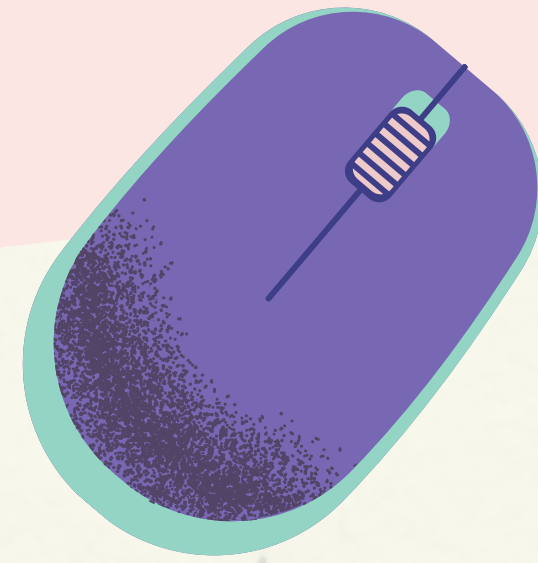
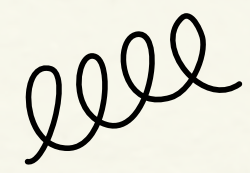
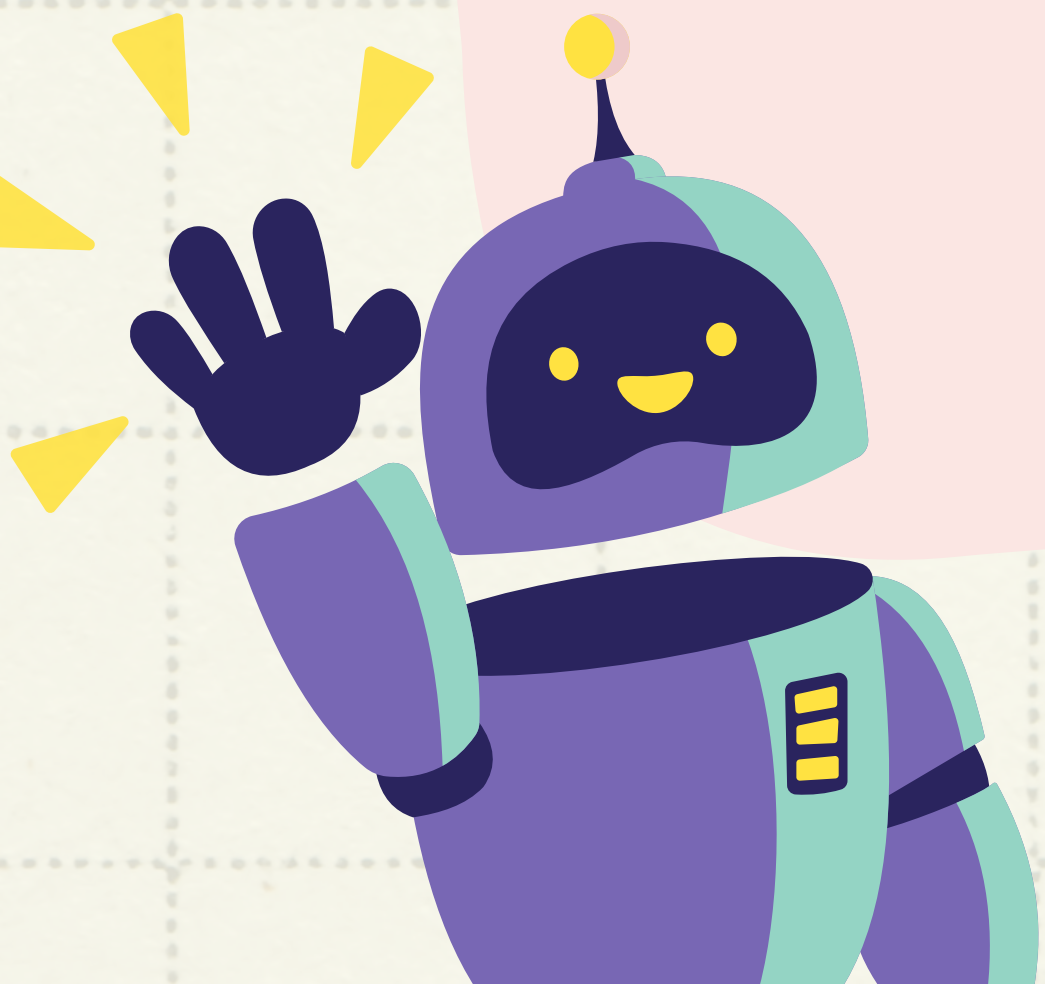
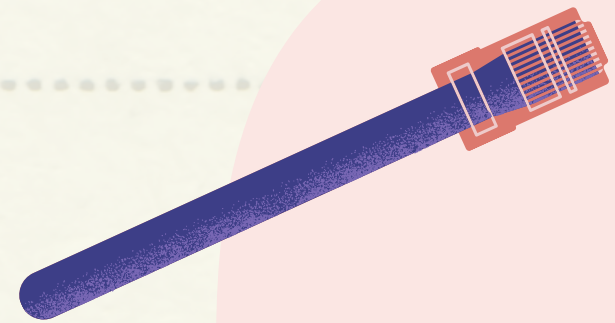


**KEEP YOURSELF  
HEALTHY!**

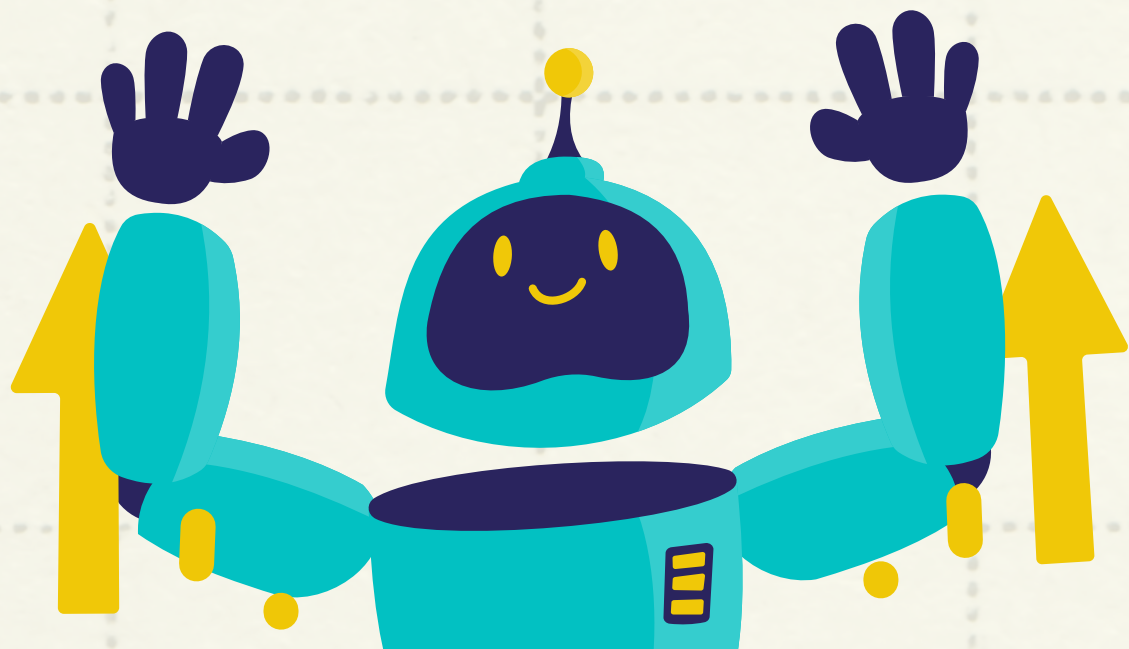




# **MEMBERS**

**NICHA RUANGRIT 6510545411**

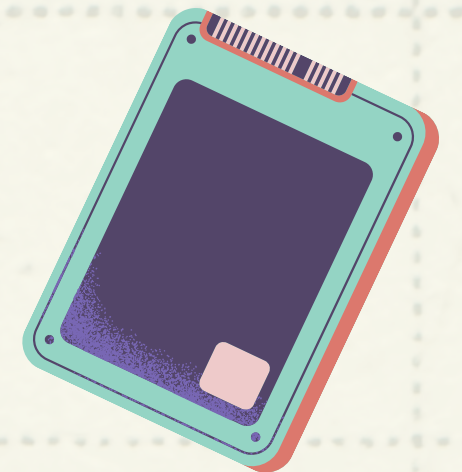
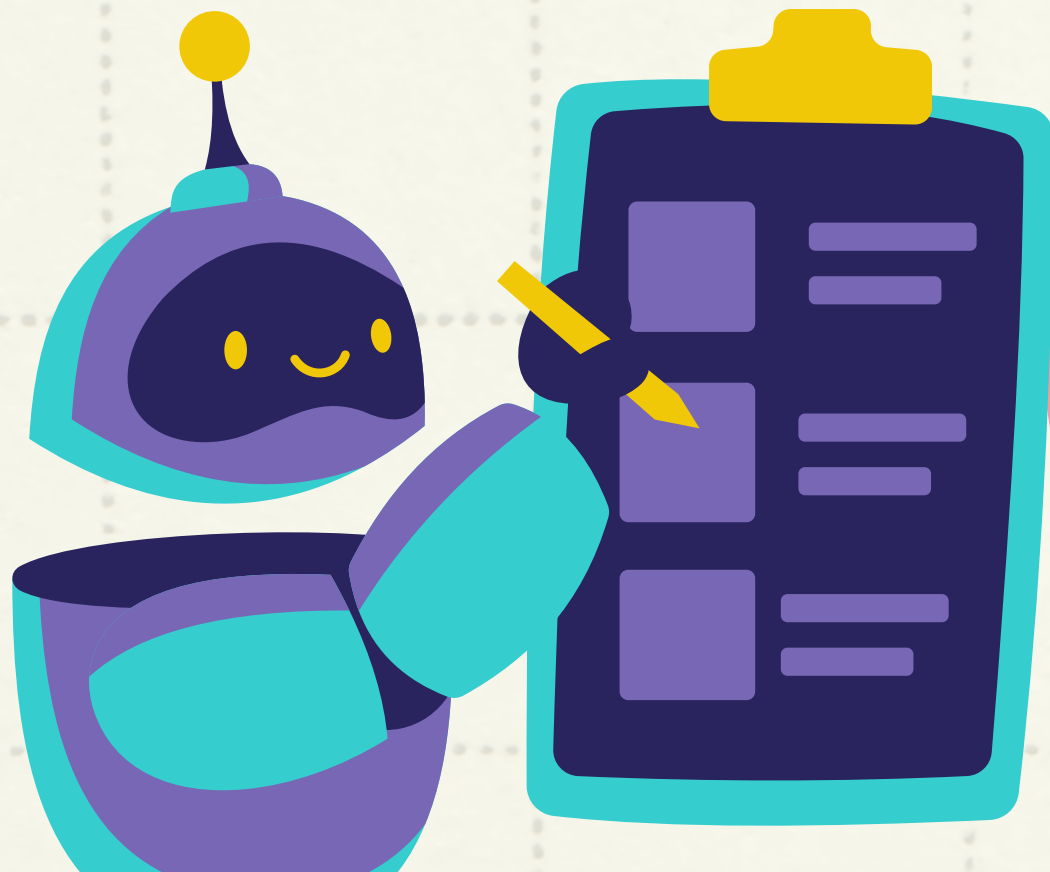
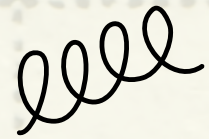
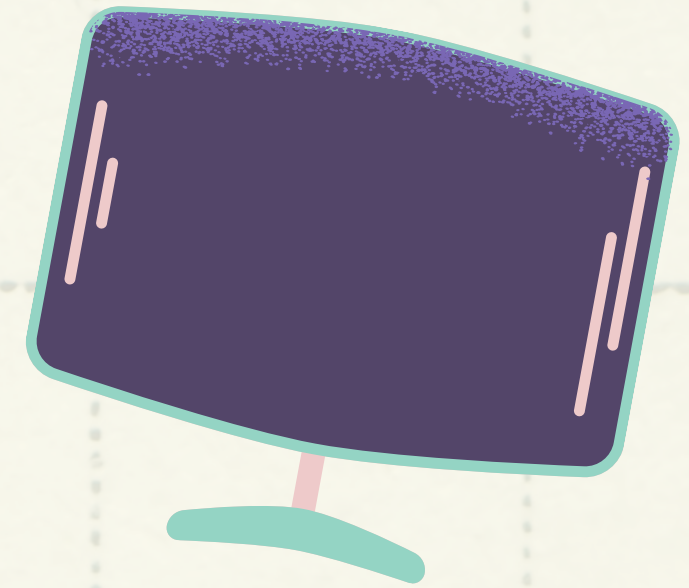
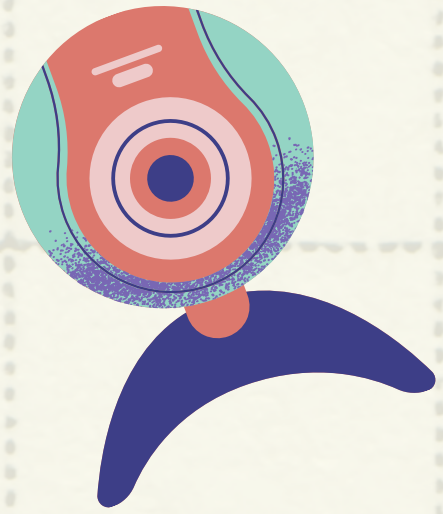
**PHAVIDA RATTANAMONGKOLKUL 6510545675**



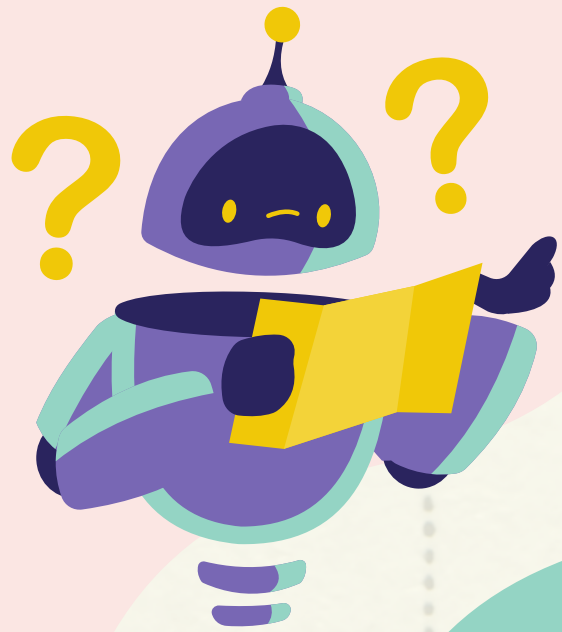


# MOTIVATION

Sitting in front of a computer screen and being still for a long time cause stress disease. So we should exercise together. But the weather on some days doesn't allow us to exercise. Due to this reason therefore we wanted to create this application for helping more people come to exercise.

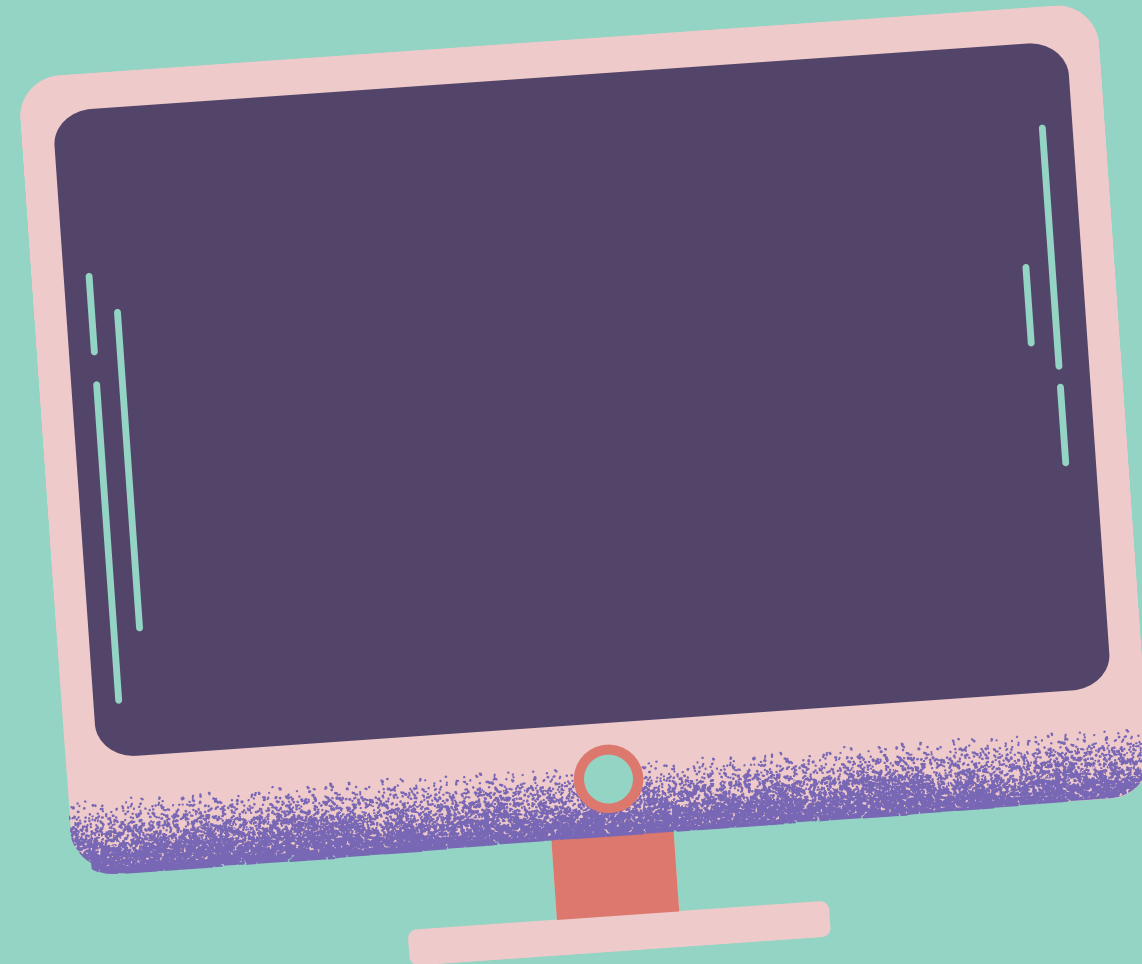






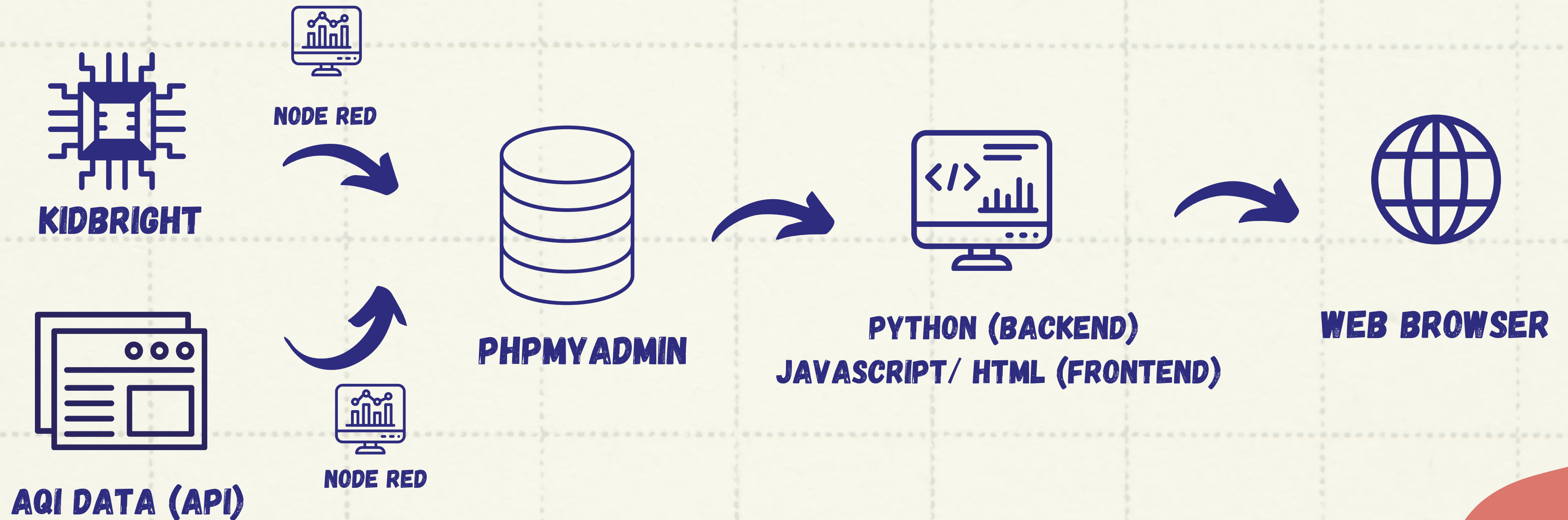
1

# OVERALL ARCHITECTURE





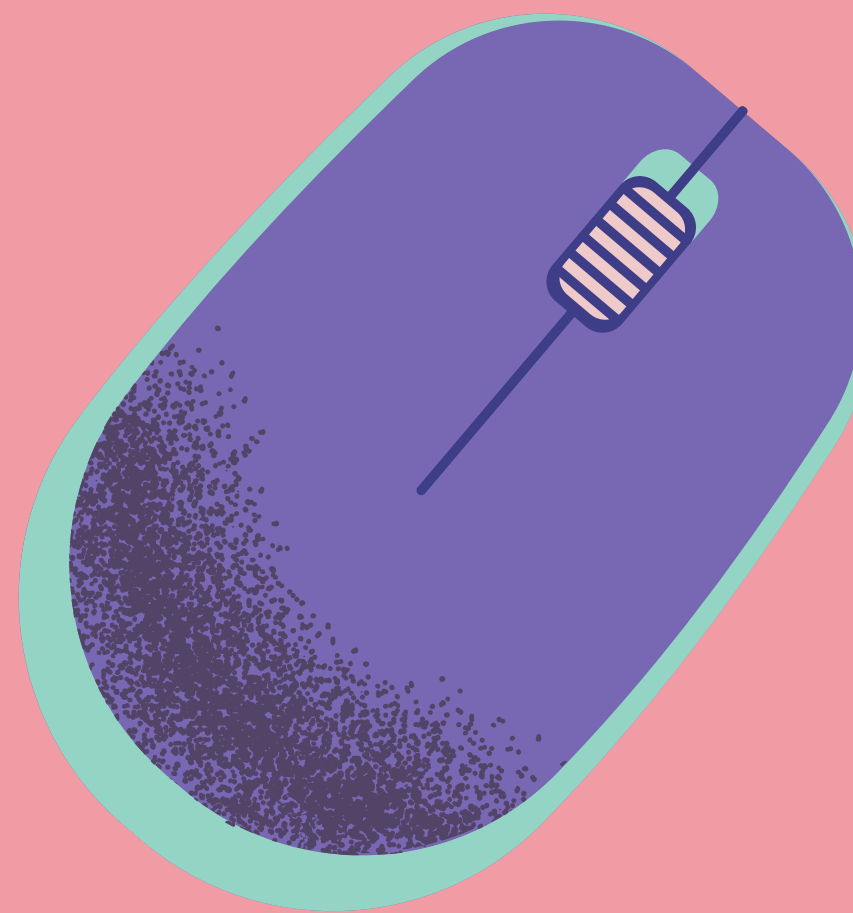
# OVERALL ARCHITECTURE





2

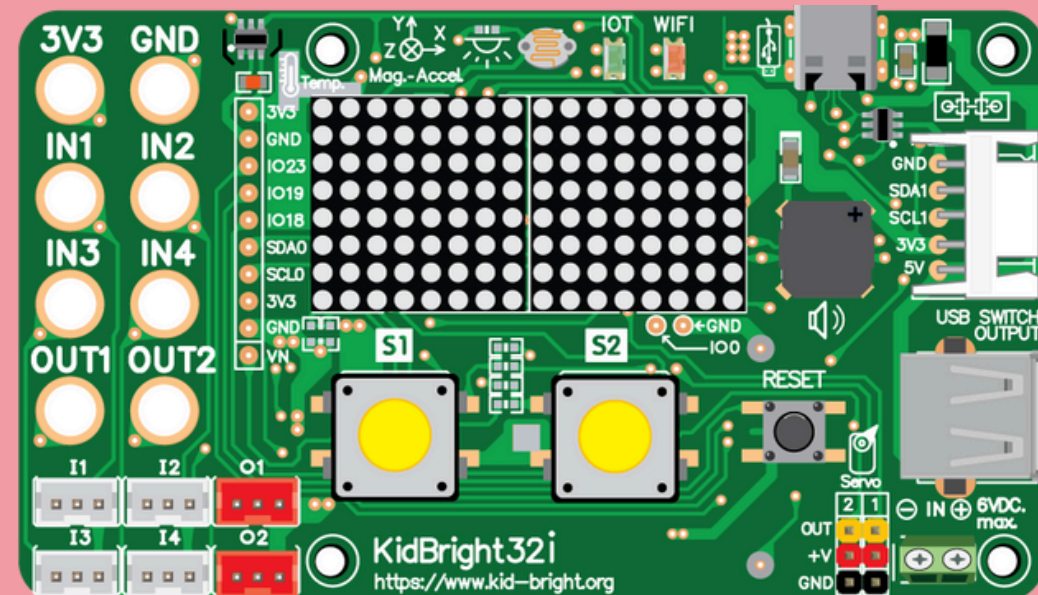
**DATA SOURCE**





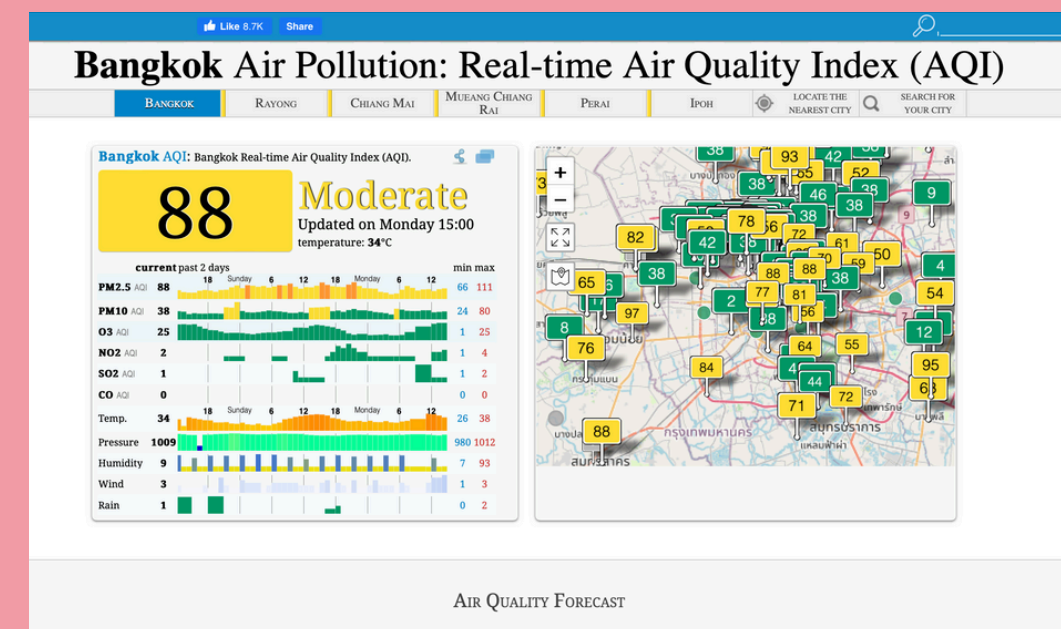
# PRIMARY

## TEMPERATURE SENSOR (KITBRIGHT)



# SECONDARY

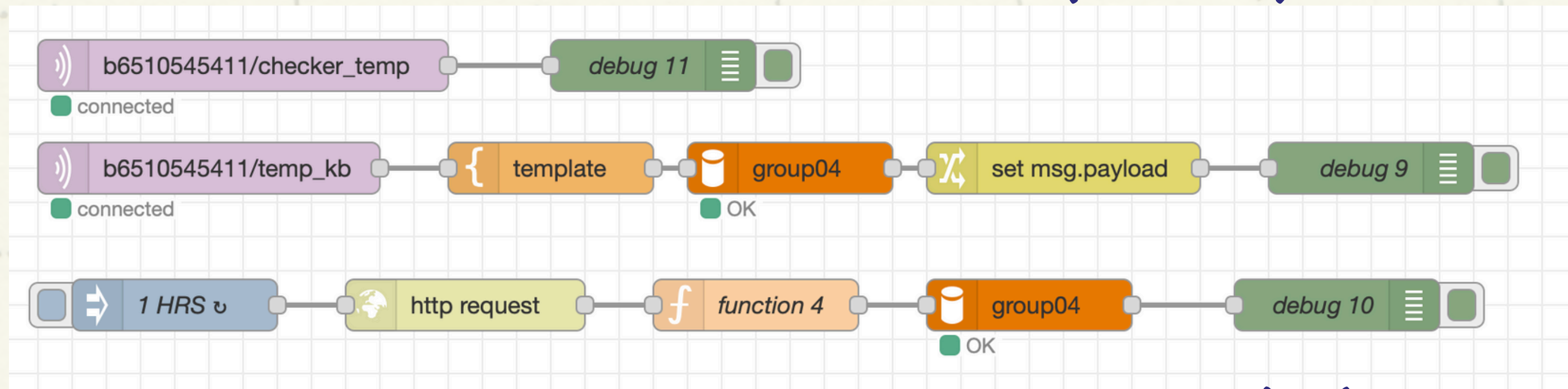
## AQI DATA (API) [HTTPS://AQICN.ORG/CITY/BANGKOK/](https://aqicn.org/city/bangkok/)





# COLLECTION MECHANISMS

## TEMPERATURE SENSOR (KITBRIGHT)



**AQI DATA (API)**  
**[HTTPS://AQICN.ORG/CITY/BANGKOK/](https://aqicn.org/city/bangkok/)**



# COLLECTION MECHANISMS

## TEMPERATURE SENSOR (KITBRIGHT)

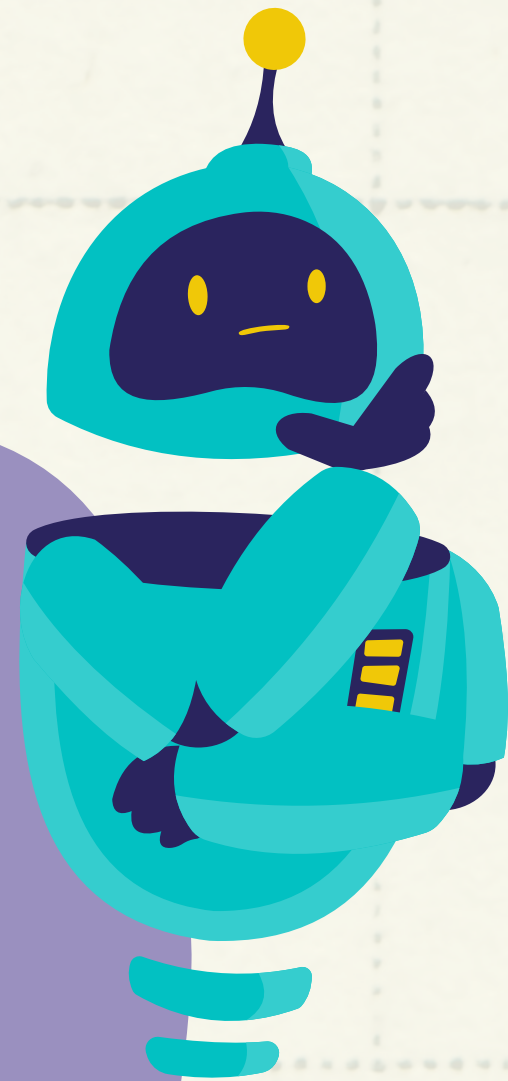
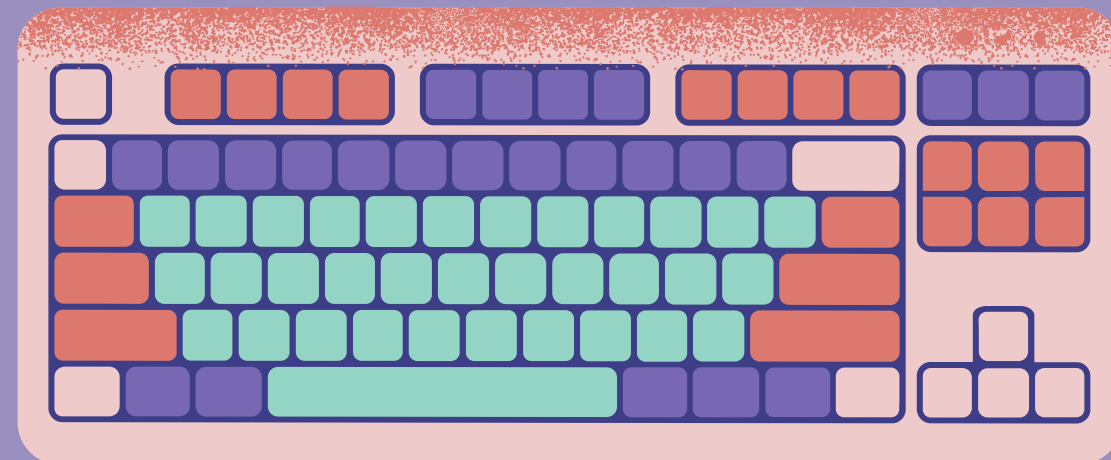
```
[ main.py ]
1  from machine import Pin, ADC, PWM, I2C
2  import network
3  import time
4  from umqtt.robust import MQTTClient
5  from config import (WIFI_SSID, WIFI_PASS, MQTT_BROKER, MQTT_USER, MQTT_PASS)
6  from math import log10
7  import json
8  import uasyncio as asyncio
9
10 PUB_MQTT_TOPIC = "b6510545411/temp_kb"
11 TOPIC_CHECKER = "b6510545411/checker_temp"
12
13 ain = ADC(Pin(36))
14 i2c = I2C(1, sda=Pin(4), scl=Pin(5))
15
16 led_wifi = Pin(2, Pin.OUT)
17 led_wifi.value(1) # turn the red led off
18 led_iot = Pin(12, Pin.OUT)
19 led_iot.value(1) # turn the green led off
20 lamp = Pin(25, Pin.OUT)
21 lamp.value(1) # turn USB lamp off initially
22
23 led_wifi = Pin(2, Pin.OUT)
24 led_wifi.value(1) # turn the red led off
25 led_iot = Pin(12, Pin.OUT)
26 led_iot.value(1) # turn the green led off
27
28 wlan = network.WLAN(network.STA_IF)
29 wlan.active(True)
30 wlan.connect(WIFI_SSID, WIFI_PASS)
31 while not wlan.isconnected():
32     time.sleep(0.5)
33 led_wifi.value(0) # turn the red led on
34 print("----> Connected to WIFI")
35
36 mqtt = MQTTClient(client_id="",
37                   server=MQTT_BROKER,
38                   user=MQTT_USER,
39                   password=MQTT_PASS)
40 mqtt.connect()
41 led_iot.value(0)
42 print("----> Connected to MQTT")
```

```
[ main.py ]
44 ain = ADC(Pin(36))
45 i2c = I2C(1, sda=Pin(4), scl=Pin(5))
46
47 def convert_i2c_to_temp():
48     i2c.scan()
49     i2c.writeto(77, bytearray([0]))
50     high, low = i2c.readfrom(77, 2)
51     hex = (high << 8) + low
52     return hex/128
53
54 async def publish_lt():
55     while True:
56         temp = convert_i2c_to_temp()
57         temp = float(temp)
58         data = {"temperature": temp}
59         print(data)
60         mqtt.publish(topic=PUB_MQTT_TOPIC, msg=json.dumps(data))
61         await asyncio.sleep(3600)
62
63 async def check_mqtt():
64     while True:
65         mqtt.check_msg()
66         await asyncio.sleep_ms(0)
67
68 async def check_alive():
69     while True:
70         await asyncio.sleep(60)
71         checker = "alive"
72         mqtt.publish(topic="b6510545411/checker_temp", msg=checker)
73         print("alive")
74
75 asyncio.create_task(publish_lt())
76 asyncio.create_task(check_mqtt())
77 asyncio.create_task(check_alive())
78 asyncio.run_until_complete()
79
```



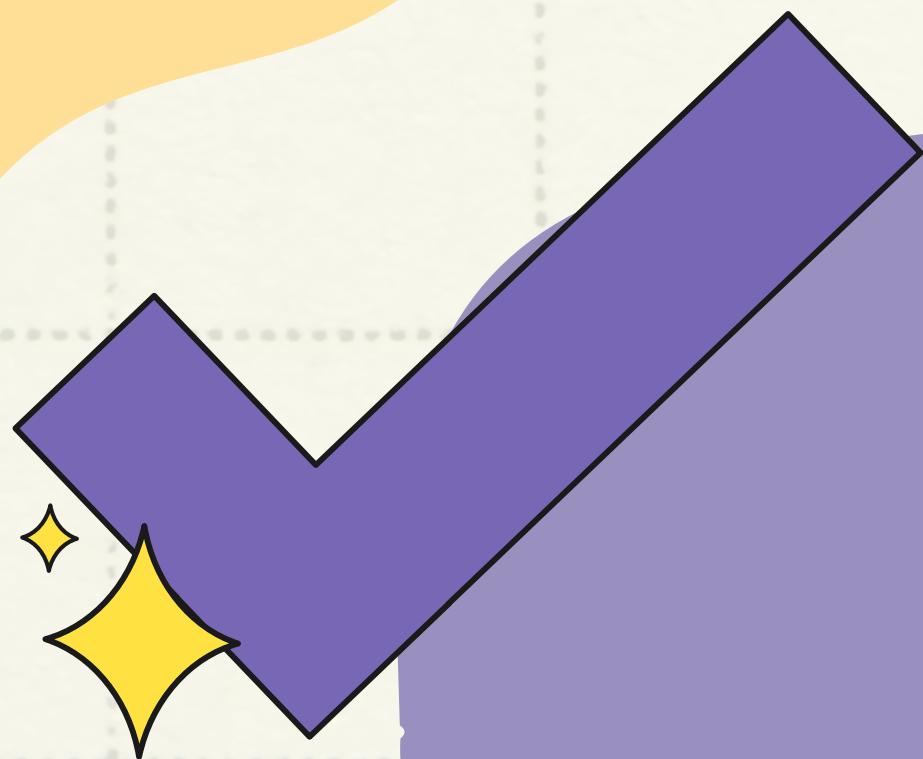
# 3


## **DATABASE SCHEMA**

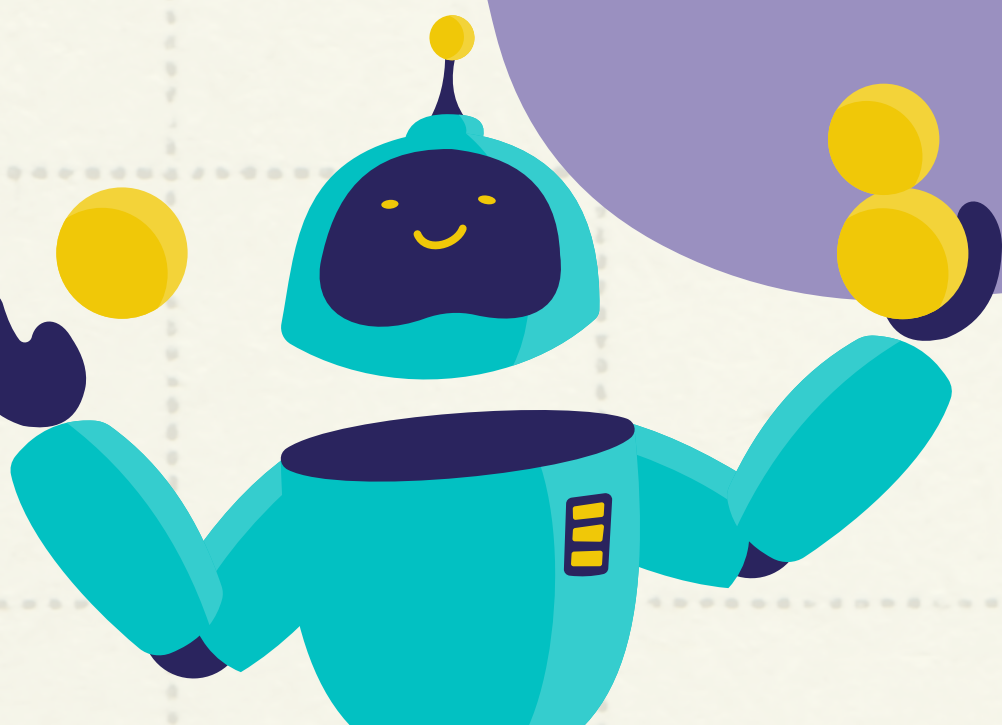




# AQI



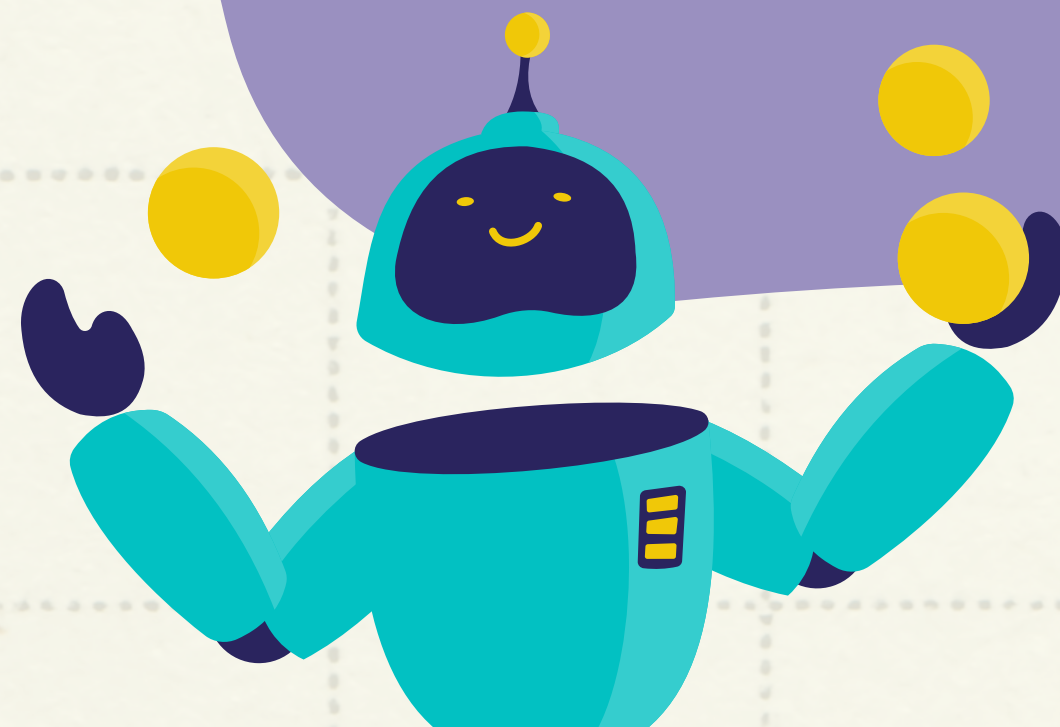
Name	Type
<u>id</u> 	int
ts	timestamp
pm25	int
location	varchar(30)





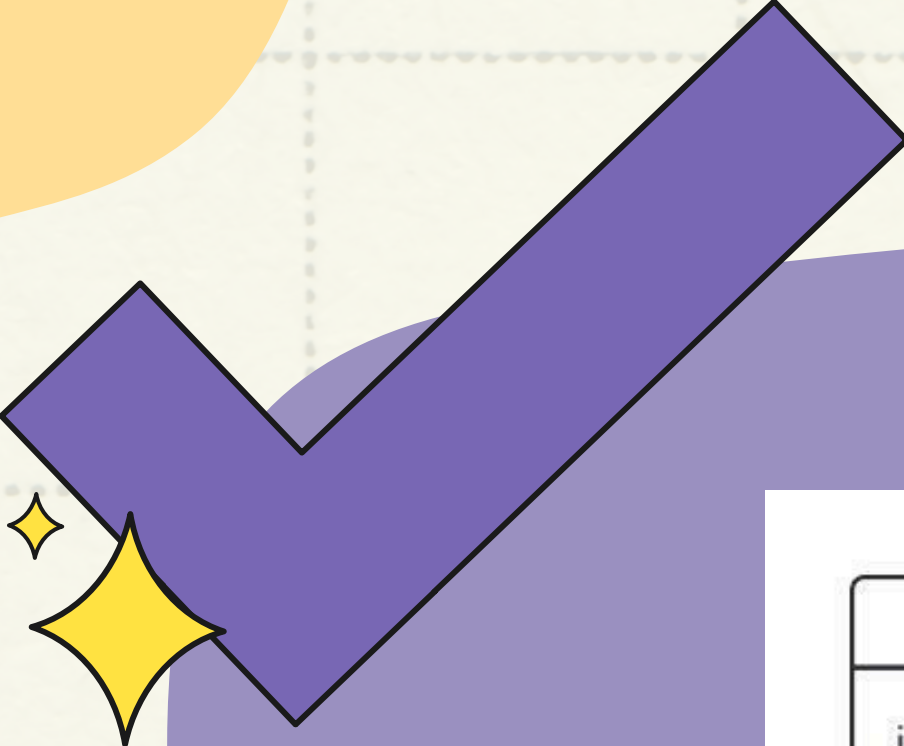
# TEMPERATURE

Name	Type
<u>id</u> 🗝️	int
ts	timestamp
temp	float
location	varchar(30)






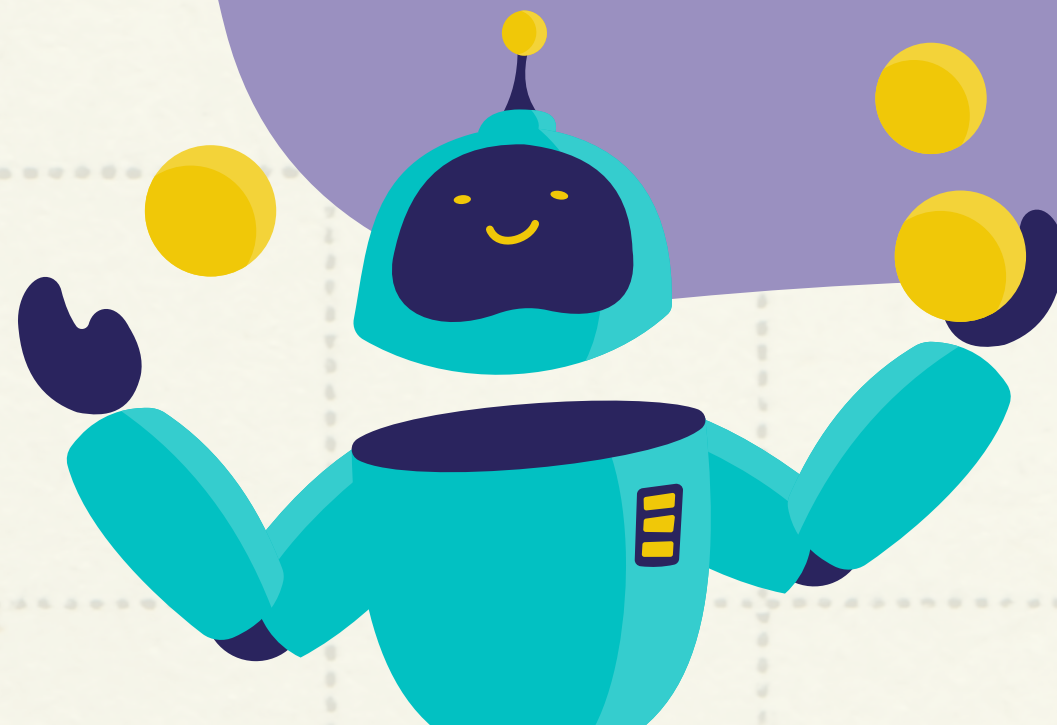
# DIAGRAM



pm25
id: int ts: timestamp pm25: int location: varchar



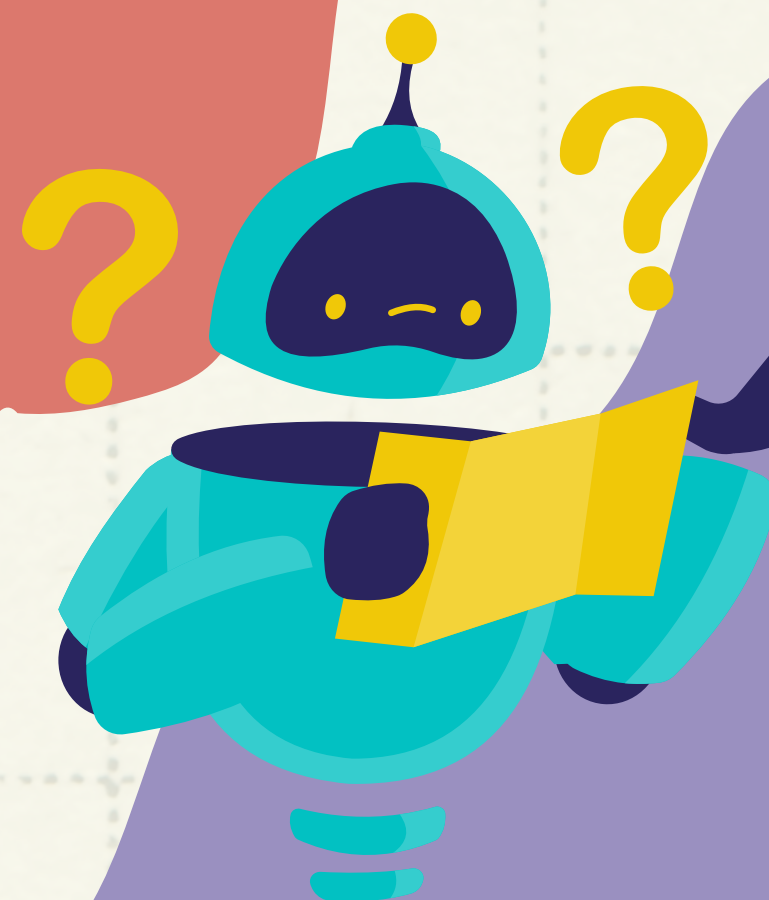
temp
id: int ts: timestamp temp: float location: varchar





4

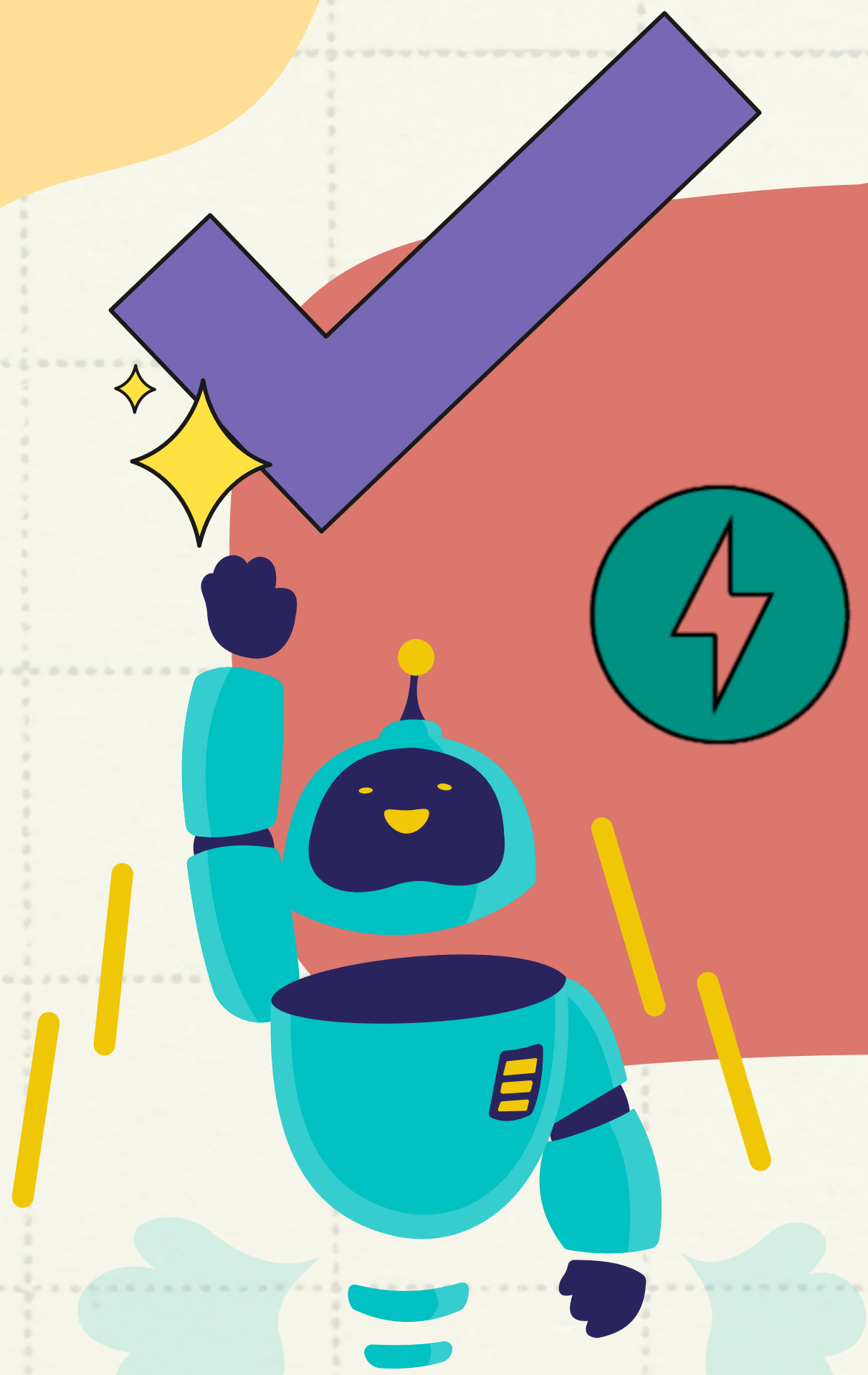
# DATA SHARING API



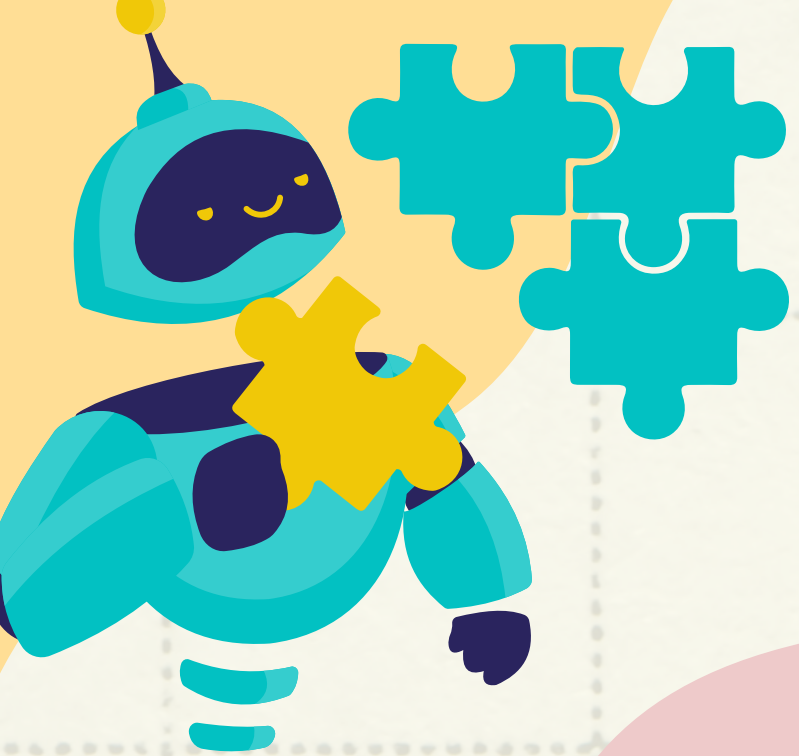




# FastAPI

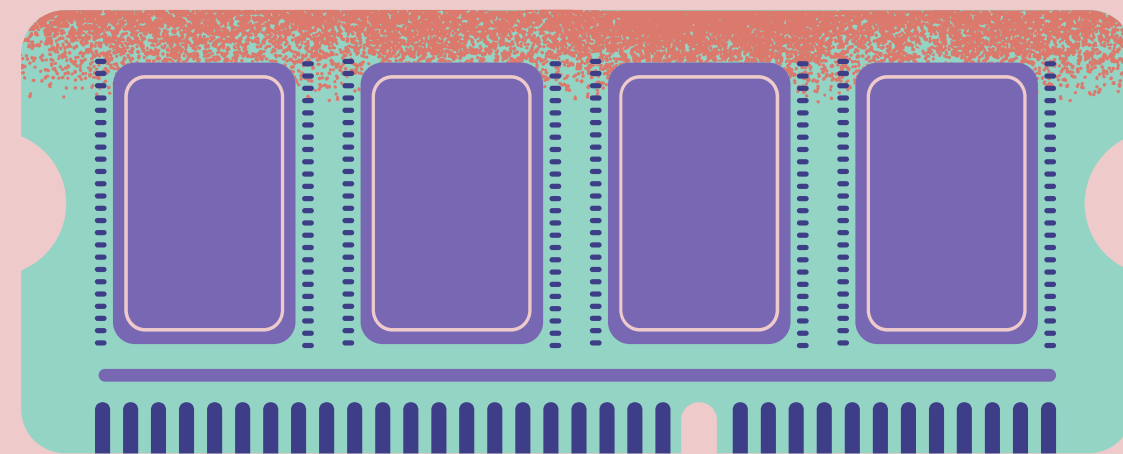






# 5

# DEMONSTRATION







**THANK YOU!**