



DAQ 2023

# GardenHolic

Presented by: **Pro-Gardener**

# Members

1. Siripa Maneein 6410545614
2. Jindaporn Sookying 6410546106
3. Kulisara Wiangin 6410545410
4. Narntatta Krivichian 6410545509





# Agenda

- **Overview**
- **Overall architecture**
- **Data sources and collection mechanisms**
- **Database schema and data integration**
- **Data sharing API**
- **Data visualization**
- **Demonstration**

# Overview

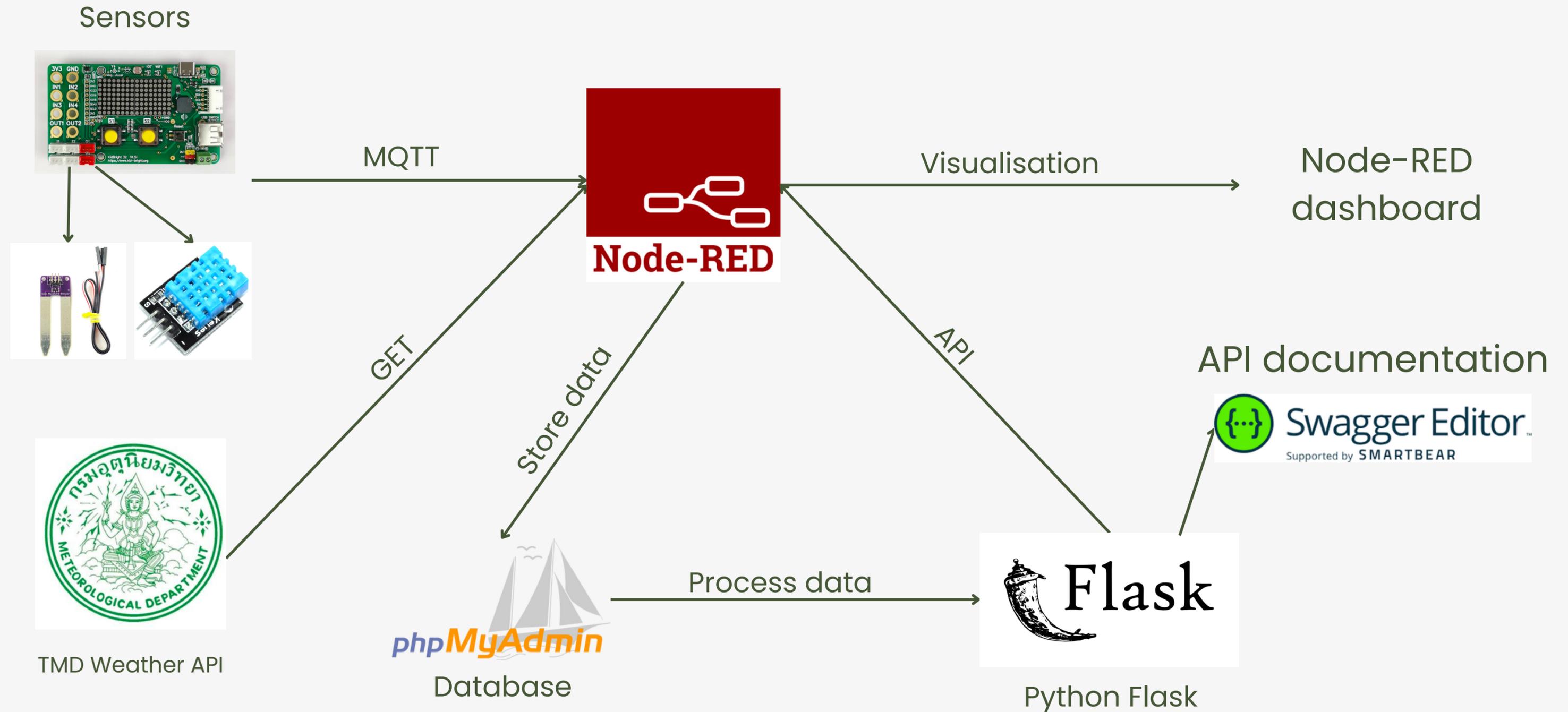
## Background

In our group, our members have many plants in the house and don't know when to water the plants. So we discovered a shared challenge: taking care of plants. In light of this, we decided to focus on creating tools that address this mutual issue.

## Motivation

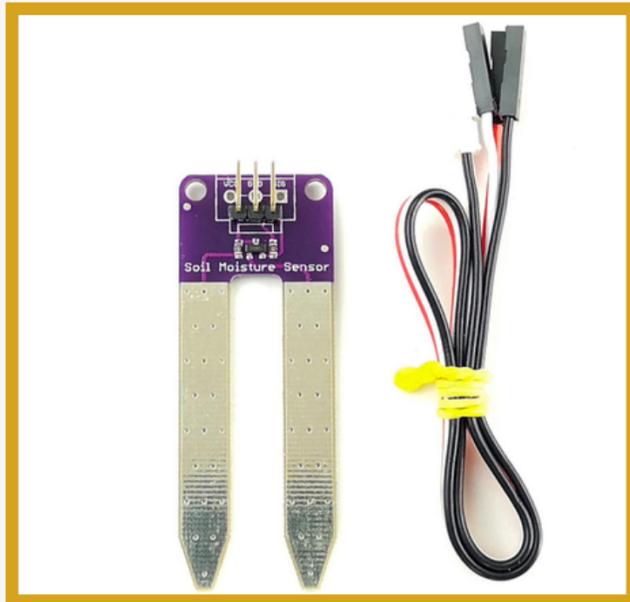
Keeping plants healthy can be a challenge, which is why we're developing a tool to determine when to water them with the use of sensors and weather prediction.

# Overall Architecture

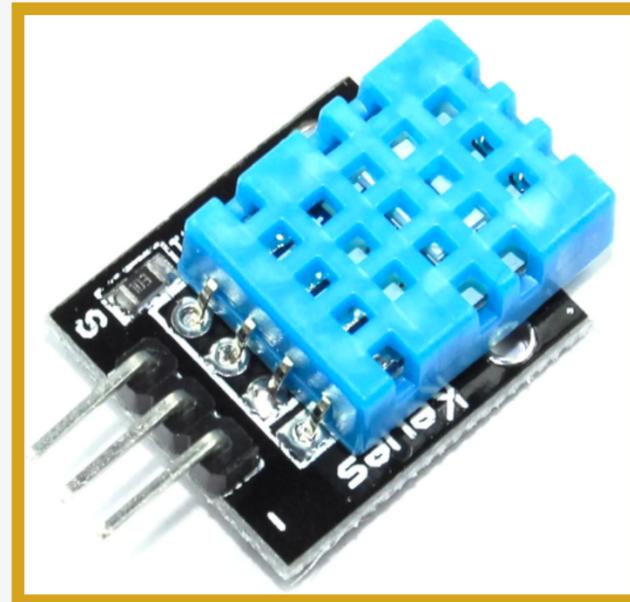


# Data Source

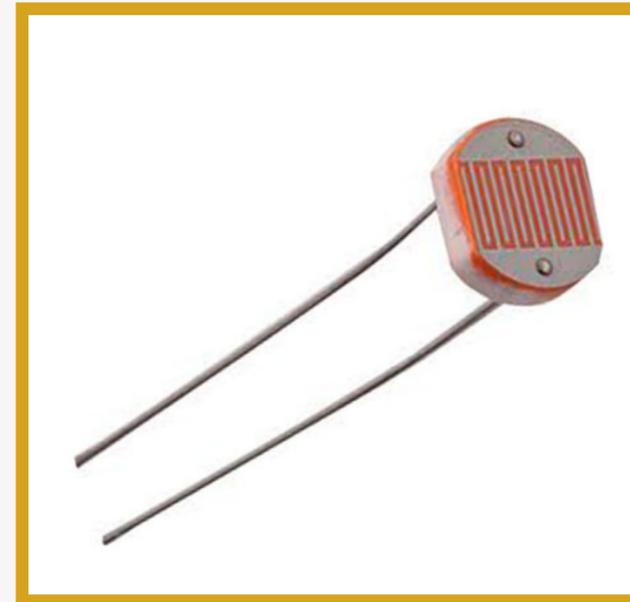
## Primary



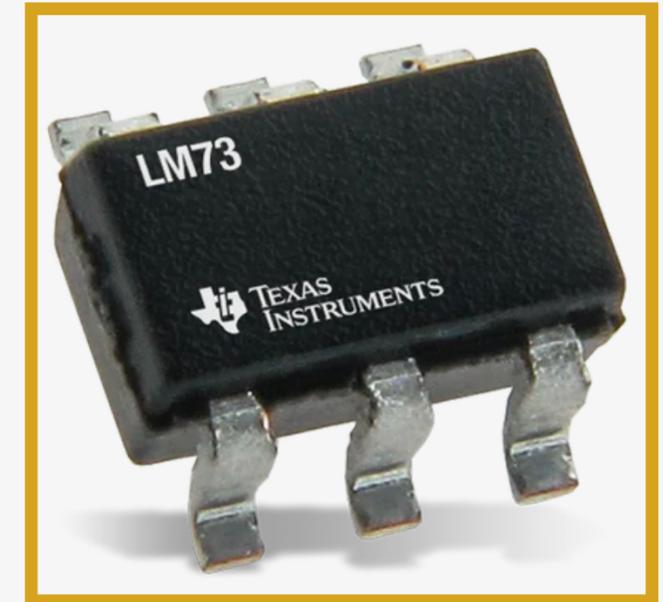
Soil moisture sensor  
(ZX-SOIL)



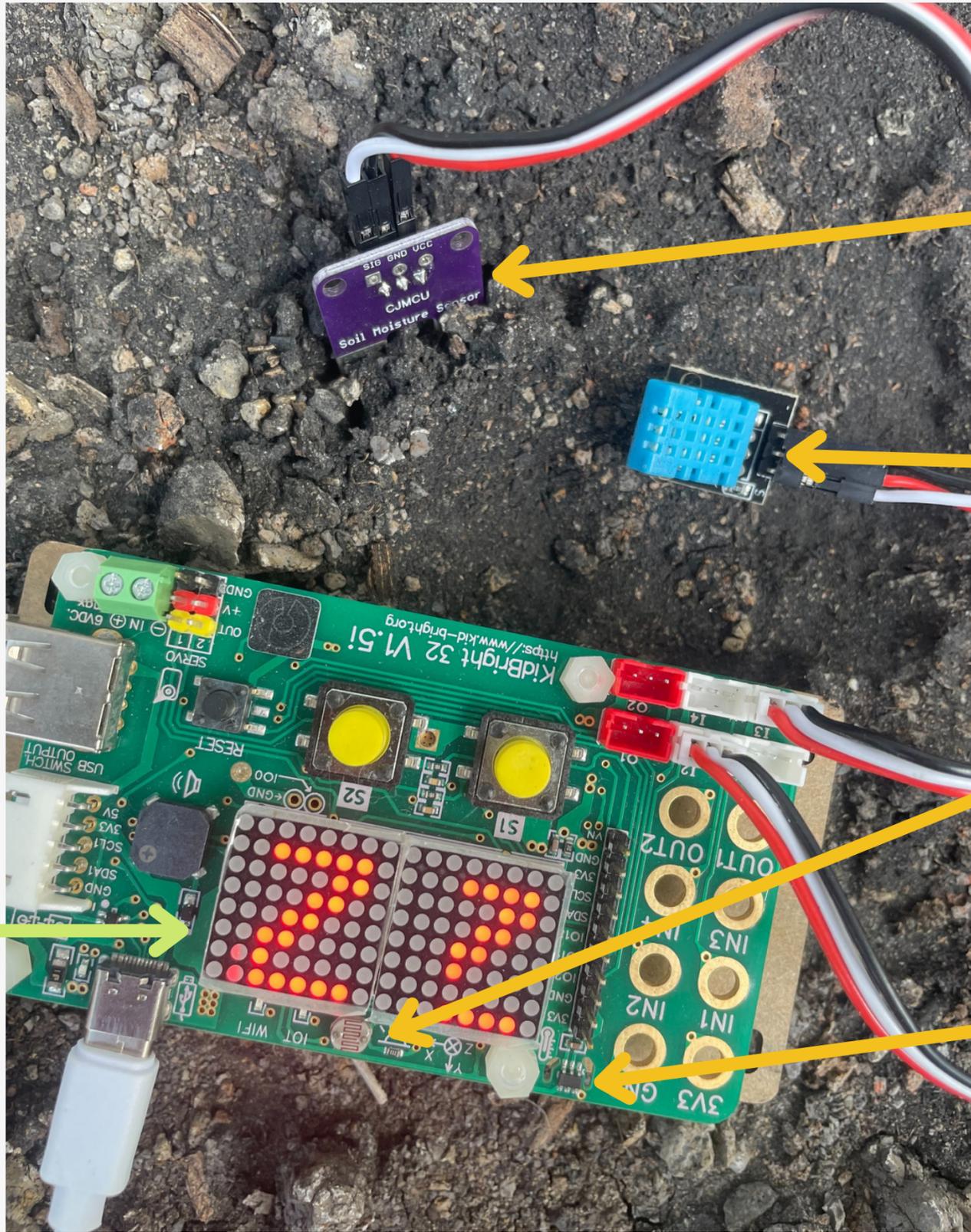
Humidity sensor  
(Temperature  
and humidity  
sensor KY-015)



Light sensor  
(Kidbright)



Temperature sensor  
(Kidbright)



Soil moisture sensor

Humidity sensor

Light sensor

Temperature sensor

LED matrix  
show kidbright status

ZZ = sleep  
OK = published data

# Data Source

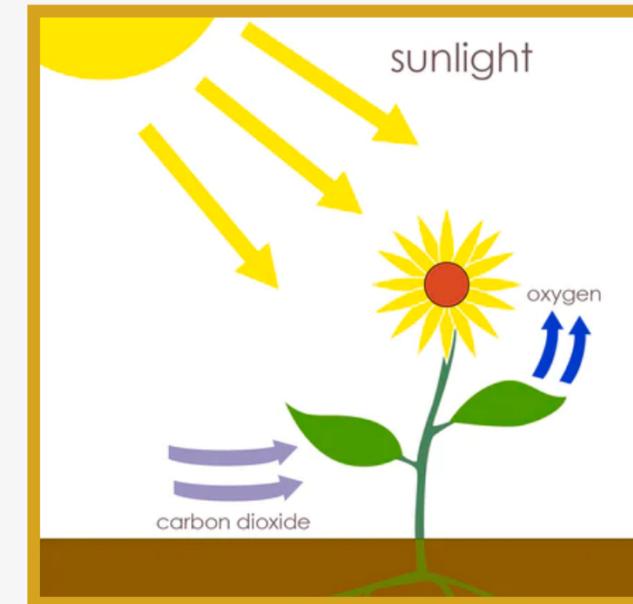
## Secondary



TMD API Forecast Daily



TMD API  
Weather3Hours



Classification of  
amount of light  
intensity in lux (low,  
medium, high

# Data Source Collection Mechanisms

**Primary data source:** Collect data from sensors every 10 minutes and publish to collected data to *daq2023/group9/* and save data to the gardener table in database in Node-RED

## Secondary data source:

- **TMD API Forecast Daily:** Collect forecast weather data from TMD every hour 2 days in advance and save to the forecast table.
- **TMD API Weather 3 Hours:** Collect actual weather data from TMD every 3 hours and save to the actual table.
- **Classification of amount of light intensity in lux:** Compare collected light and show light level in UI
  - **Low:** less than 807 Lux
  - **Medium:** 807 - 1614 Lux
  - **High:** more than 1614 Lux



# Database Schema

Name	Type
<b>id</b> 	int(11)
<b>ts</b>	timestamp
<b>lat</b>	float
<b>lon</b>	float
<b>humid</b>	float
<b>temp</b>	float

**actual**

Name	Type
<b>id</b> 	int(11)
<b>ts</b>	timestamp
<b>lat</b>	float
<b>lon</b>	float
<b>humid</b>	float
<b>temp</b>	float

**forecast**

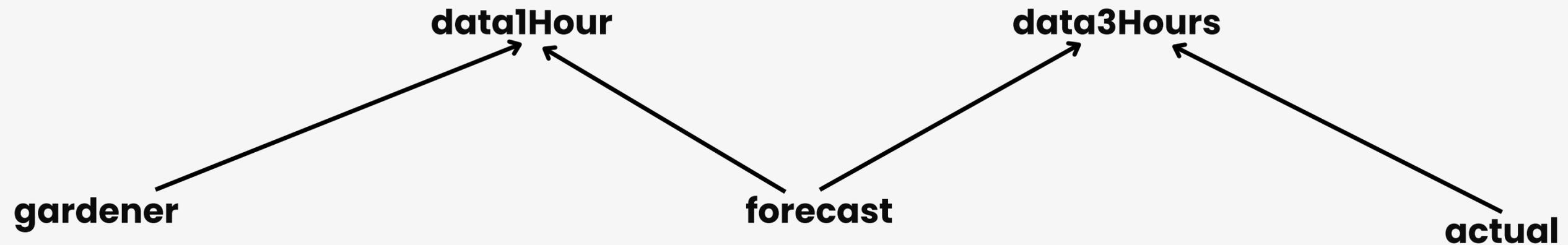
Name	Type
<b>id</b> 	int(11)
<b>ts</b>	timestamp
<b>lat</b>	float
<b>lon</b>	float
<b>soil</b>	float
<b>humid</b>	float
<b>temp</b>	float
<b>light</b>	float

**gardener**

# Database Schema and Data Integration

Name	Type
id 	int(11)
ts	timestamp
lat	float
lon	float
avg_temp	float
avg_humid	float
source	varchar(255)

Name	Type
id 	int(11)
ts	timestamp
lat	float
lon	float
avg_temp	float
avg_humid	float
source	varchar(255)





# DATA SHARING API

<https://github.com/Siripa-Maneein/gardenholic>

# Data sharing API

## **/latest\_sensor**

Return the latest sensor data from kidbright.

### **Response example:**

```
{  
  "humidity": 57,  
  "lat": 13.5795,  
  "light": 165.523,  
  "lon": 100.593,  
  "soil": 0,  
  "temperature": 27.25,  
  "time": "2023-11-22T10:00:00Z"  
}
```



# Data sharing API

## `/sensors_hourly/{duration}`

Show time, lat, lon, soil, humidity, temperature, light data from kidbright source in each hour for the past given {duration} days.

### Response example:

```
[  
  {  
    "humidity": 57,  
    "lat": 13.5795,  
    "light": 165.523,  
    "lon": 100.593,  
    "soil": 0,  
    "temperature": 27.25,  
    "time": "2023-11-22T10:00:00Z"  
  },  
  {  
    "humidity": 57,  
    "lat": 13.5795,  
    "light": 165.523,  
    "lon": 100.593,  
    "soil": 0,  
    "temperature": 27.25,  
    "time": "2023-11-22T11:00:00Z"  
  },  
  ...  
]
```



# Data sharing API

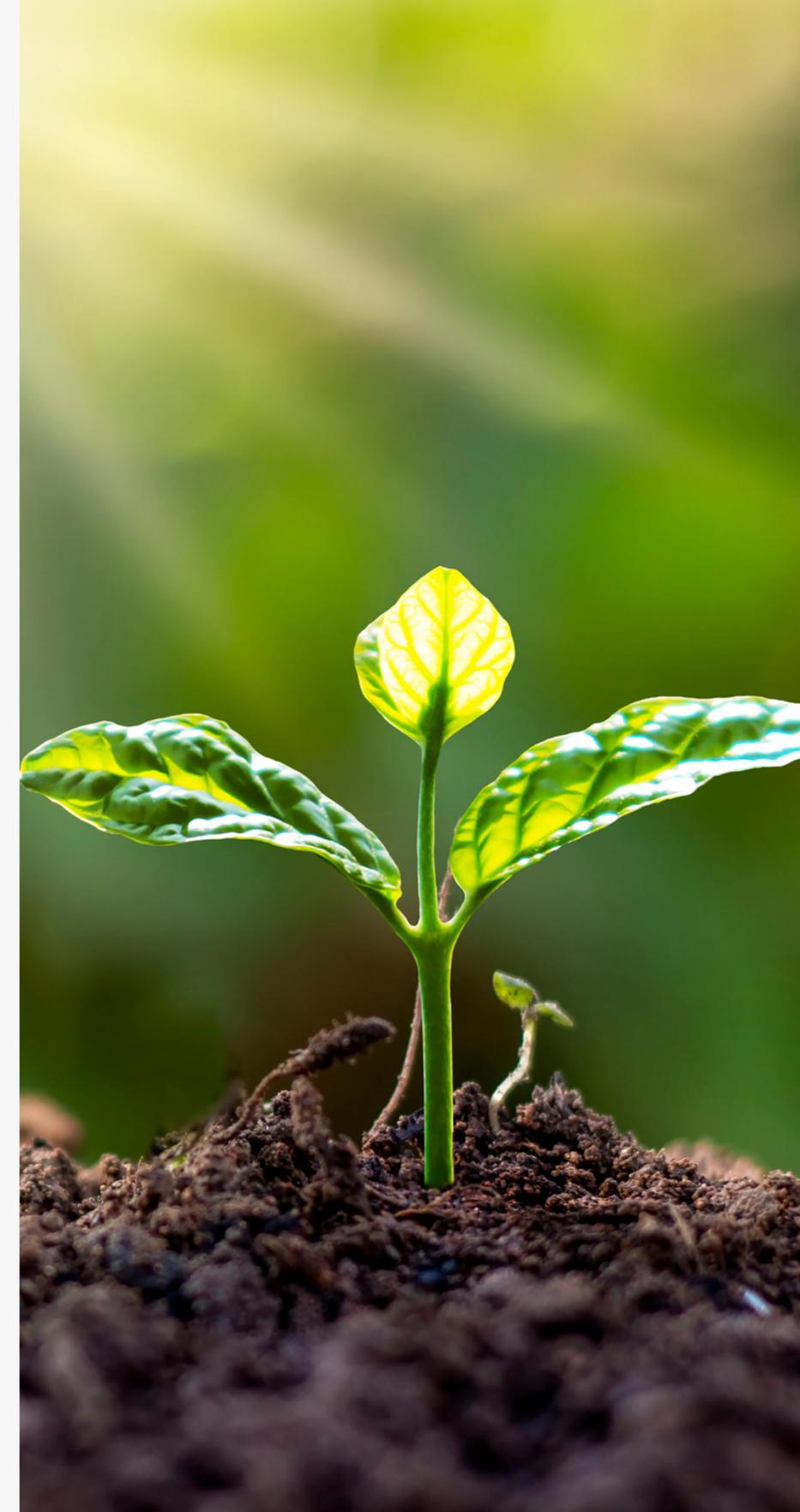
## **/sensors\_hourly\_by\_date/{date}**

Show time, lat, lon, soil, humidity, temperature, light data from kidbright source in each hour in a specific date (YYYY-MM-DD)

ex. /sensors\_hourly/2023-11-22

### **Response example:**

```
[
{
  "humidity": 57,
  "lat": 13.5795,
  "light": 165.523,
  "lon": 100.593,
  "soil": 0,
  "temperature": 27.25,
  "time": "2023-11-22T00:00:00Z"
}, ...
{
  "humidity": 57,
  "lat": 13.5795,
  "light": 165.523,
  "lon": 100.593,
  "soil": 0,
  "temperature": 27.25,
  "time": "2023-11-22T23:00:00Z"
}
]
```



# Data sharing API

## `/should_i_water_my_plant`

Check the forecast api and your plant humidity

```
if soil moisture < 50%:  
    if in 24 hours, it will rain (usually humidity higher than 90%):  
        return "don't water the plant" aka "False"  
    return "water the plant" aka "True"  
return "don't water the plant" aka "False"
```

### Response Example:

```
{water: True}
```



# Data sharing API

## **/forecast\_data**

Return forecast data collected from TMD for the past 3 days and onward

### Response example:

```
[
  {
    "humidity": 55.89,
    "lat": 13.9208,
    "lon": 100.609,
    "temperature": 26.06,
    "time": "2023-11-25T22:00:00Z"
  },
  {
    "humidity": 58.68,
    "lat": 13.9208,
    "lon": 100.609,
    "temperature": 25.67,
    "time": "2023-11-25T23:00:00Z"
  },
  ...
]
```



# Data sharing API

## **/actual\_data**

Return actual data collected from TMD for the past 3 days

### **Response example:**

```
[
  {
    "humidity": 62.0,
    "lat": 13.9192,
    "lon": 100.605,
    "temperature": 28.3,
    "time": "2023-11-25T22:00:00Z"
  },
  {
    "humidity": 62.0,
    "lat": 13.9192,
    "lon": 100.605,
    "temperature": 27.0,
    "time": "2023-11-26T01:00:00Z"
  },
  ...
]
```



# Data sharing API

## **/tmd\_accuracy**

Compare forecast api and actual api at the same hour for the past 3 days

- The forecast data need to be grouped into 3-hour intervals first.
- Use MAPE forecast accuracy (Mean Absolute Percentage Error) to compare
- Calculate the %error between values at the same interval of the past 3 days  
 $\%error = \text{diff}(\text{forecast} - \text{actual}) / \text{actual} * 100$
- Find Average of %Error  
 $\%AverageError = \text{sum}(\%error) / \text{number\_of\_rows}$
- Find %Accuracy  
 $\%Accuracy = 100 - \%AverageError$

### **Response example:**

```
{  
  "humidity_accuracy_percentage": 92,  
  "temperature_accuracy_percentage": 90,  
}
```



# Data sharing API

## **/tmd\_accuracy/sensors**

Compare forecast api and sensors data collected at the same hour for the past 3 days

- The sensors data need to be grouped into 1-hour interval first.
- Silimar to previous but change the compared data to the sensors that we collect

### **Response example:**

```
{  
  "humidity_accuracy_percentage": 90,  
  "temperature_accuracy_percentage": 89,  
}
```



# Data Visualization

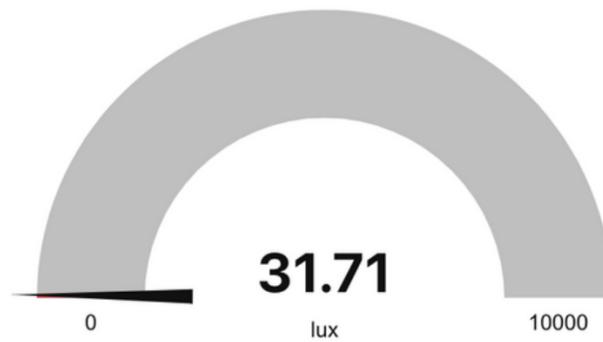
☰ Current data

## Current sensors data

### Soil Moisture

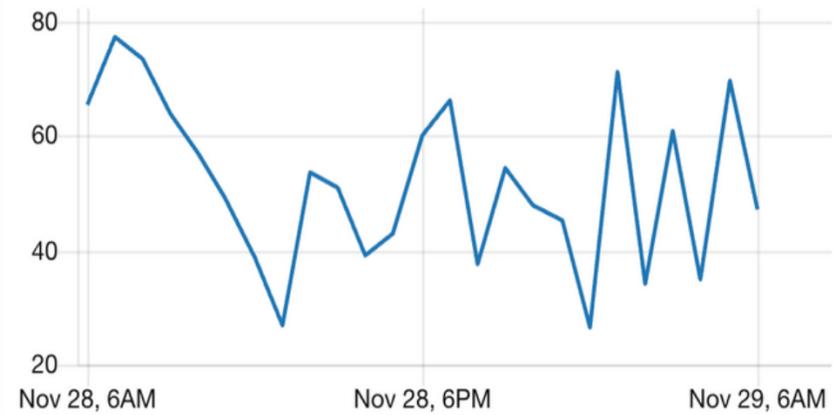


### Light Intensity

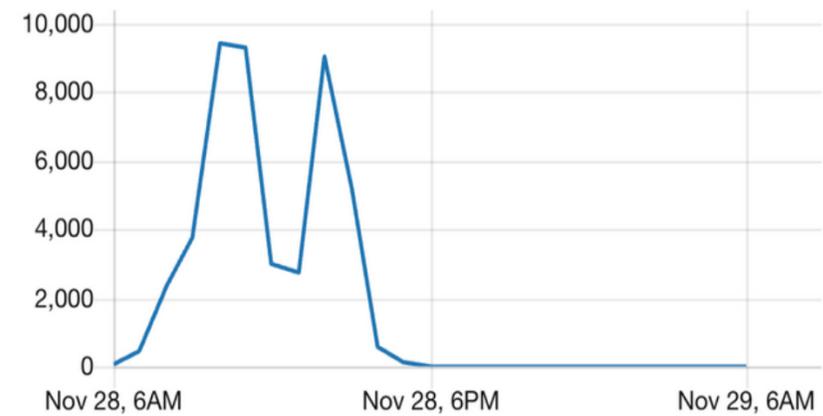


## Sensors data chart

### Soil Moisture



### Light Intensity



## Should I water my plant

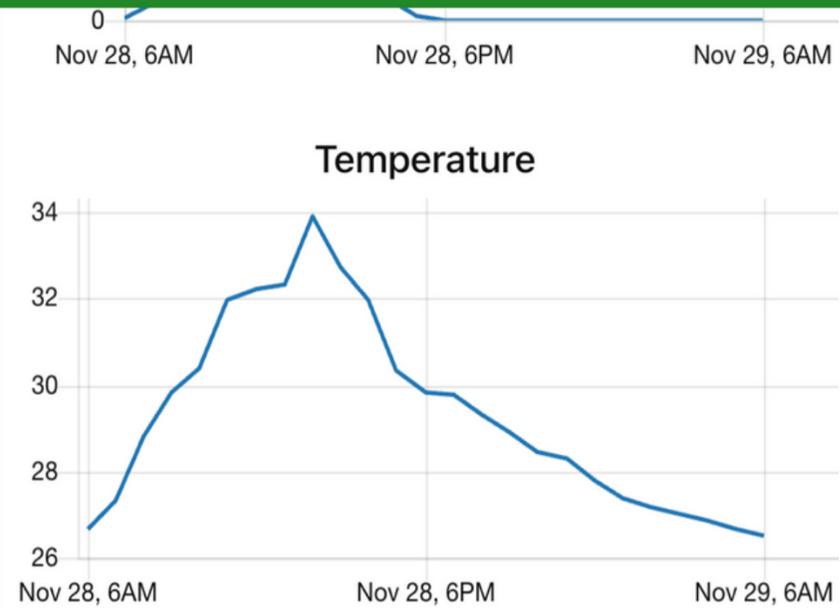
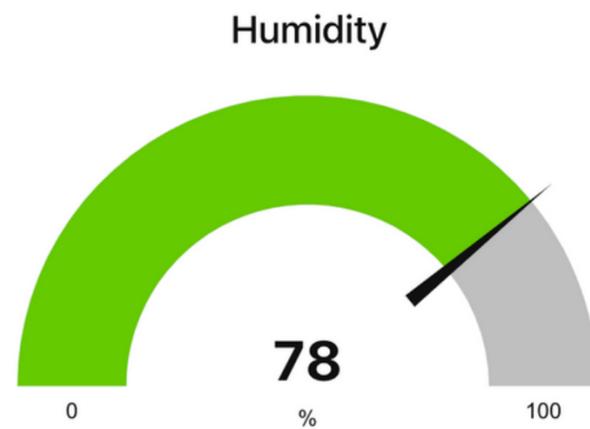
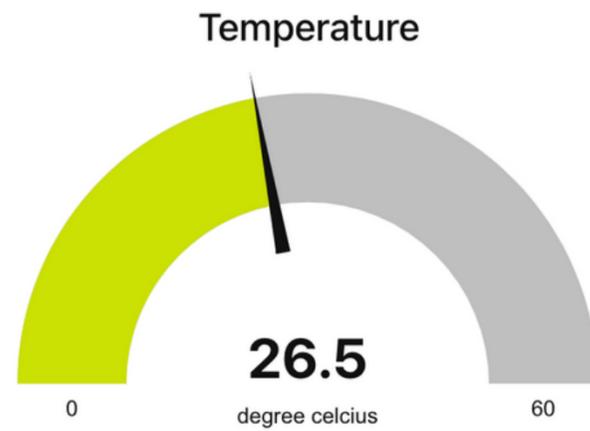
No

## Level of Light Intensity

Low

# Data Visualization

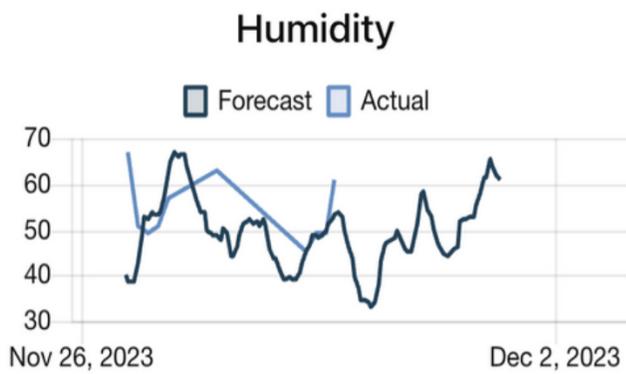
☰ Current data



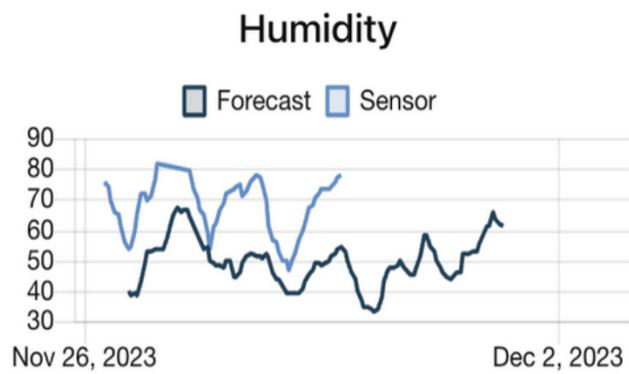
# Data Visualization

☰ TMD Comparison

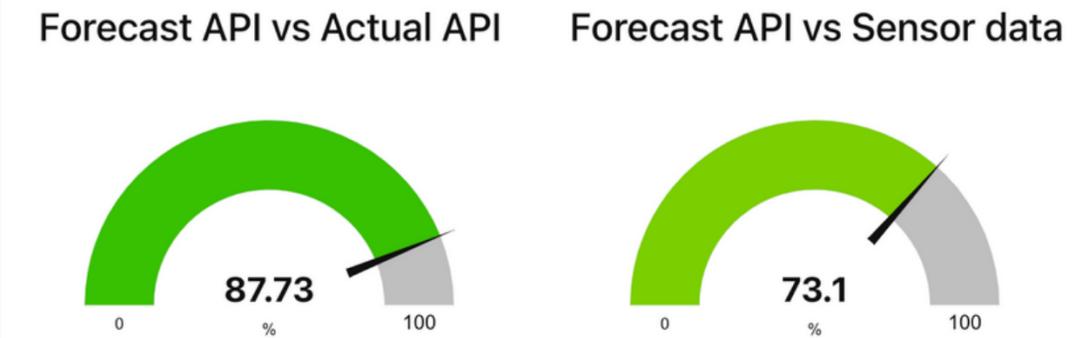
### Forecast API vs Actual API



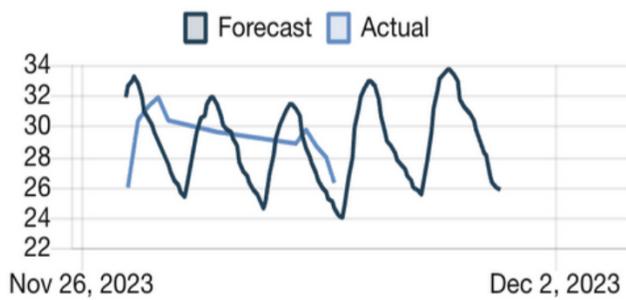
### Forecast API vs Sensor



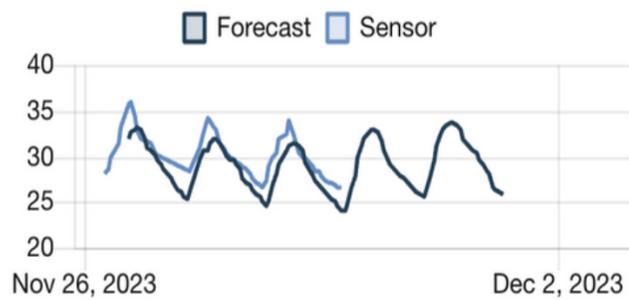
### Humidity Accuracy



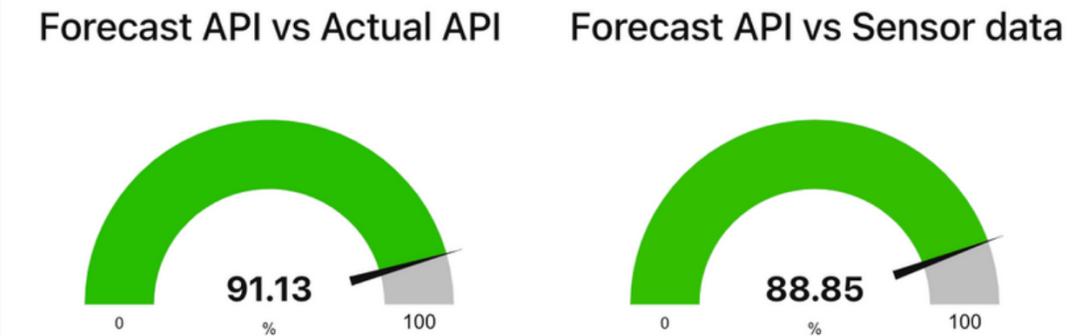
### Temperature



### Temperature



### Temperature Accuracy



# Demo





**Thank You**

