



# HomieCare

**Monitor indoor conditions + IP Camera with CV tracks actions  
integrates with weather data for advice, and notifies via LINE in emergencies.**

---

**Sirin PHUNGKUN**  
**Thanabordee Bunditsakul**

**6510545730**  
**6510545489**



# Background

---

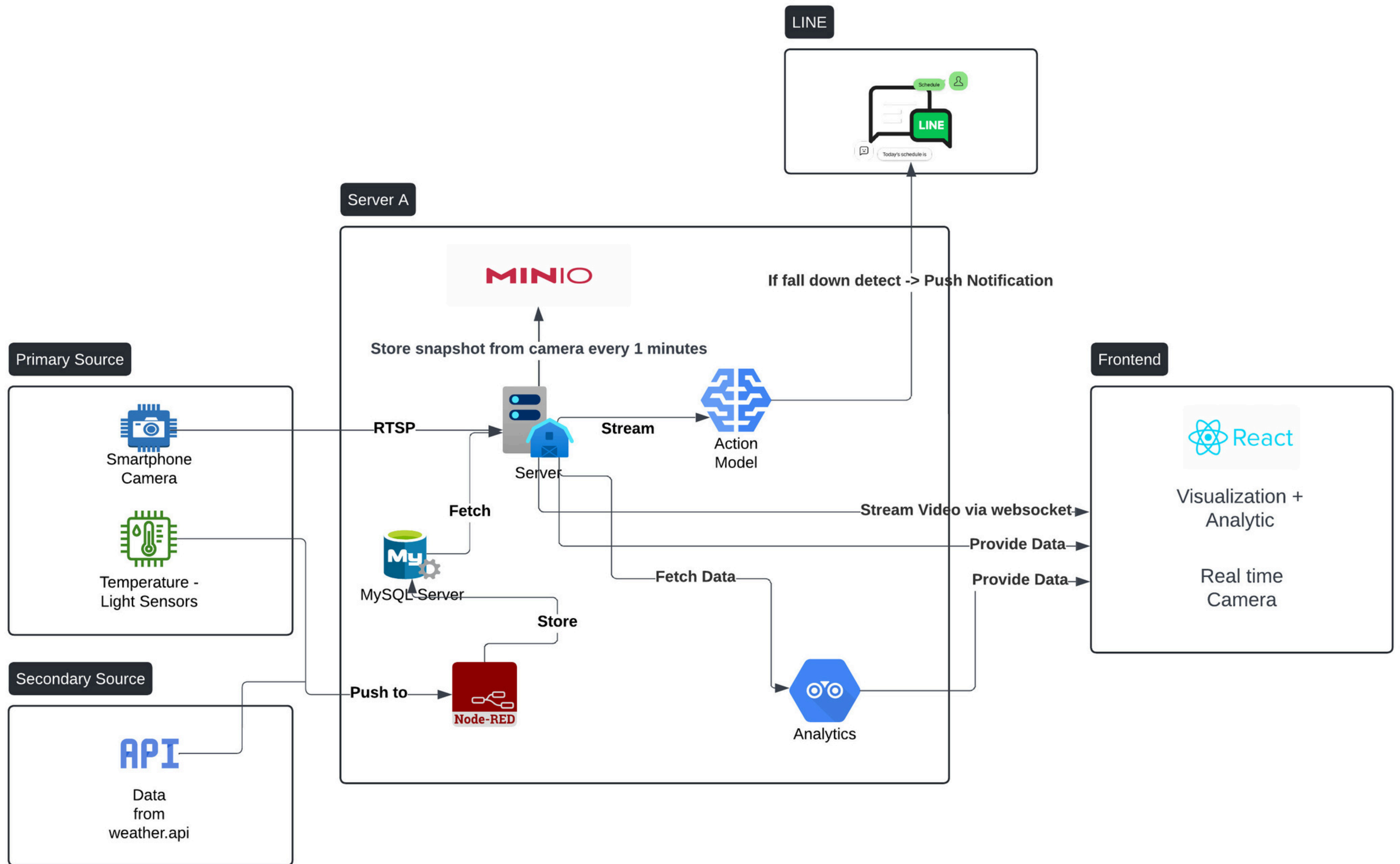
**A system to watch over people living alone. It uses an ESP32 with sensors to track indoor environments such as temperature and light + IP Camera with an action model to keep track of action status. Combines this information with other online sources, like weather data, to give advice and notify us when bad events happen.**



The background features a dark blue field with intricate, glowing magenta geometric patterns. These patterns consist of numerous thin, parallel lines that form a series of nested, slightly offset rectangular frames, creating a sense of depth and architectural structure. The lines are most concentrated on the right side of the image, where they form a dense, tunnel-like effect, and become more sparse towards the left.

# **System *Architecture***









# **Data source and Collection Mechanism**

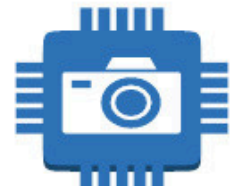


# Data source : Primary

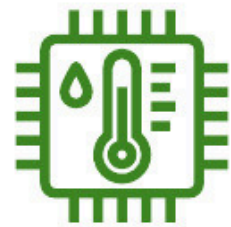
---

Place both source indoor

Primary Source



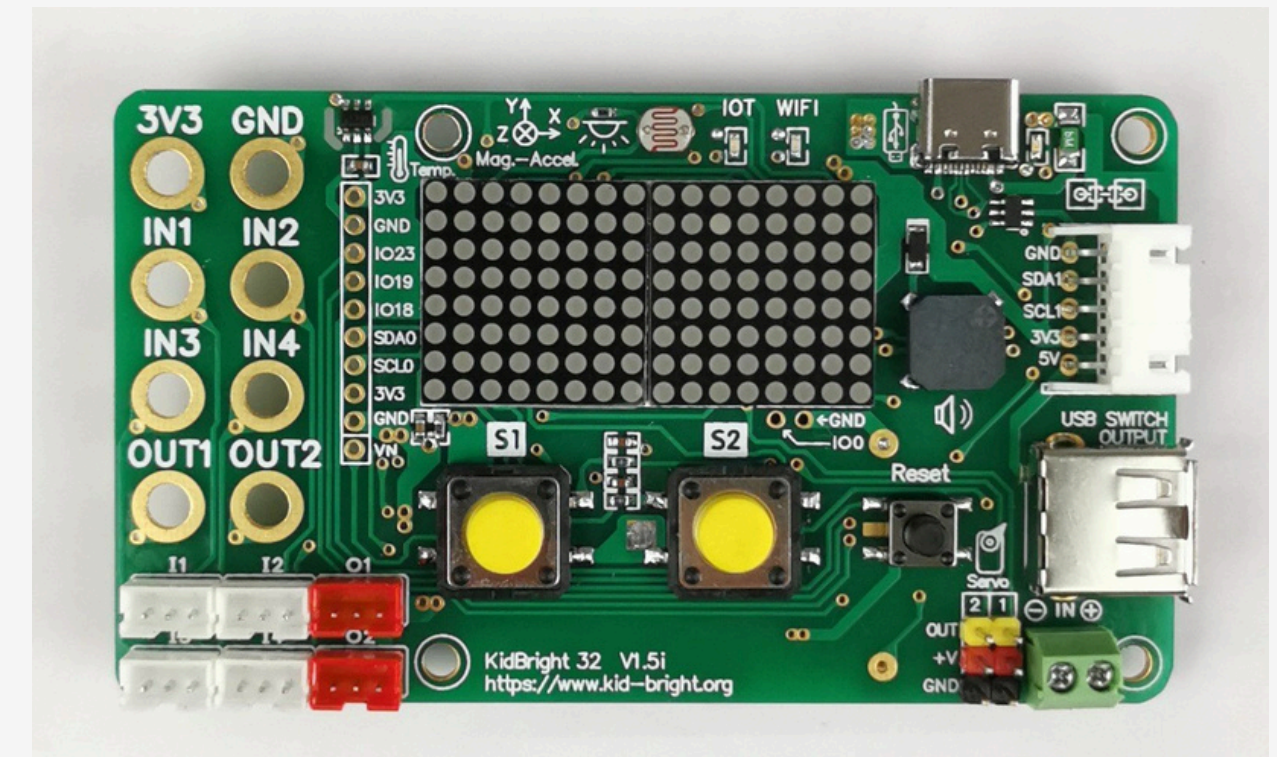
Smartphone  
Camera



Temperature -  
Light Sensors



**Samsung Galaxy A12**  
+  
**IP Camera Application**



**Kidbright**  
**Temperature and Light**



# Primary source : Collection Mechanism

---

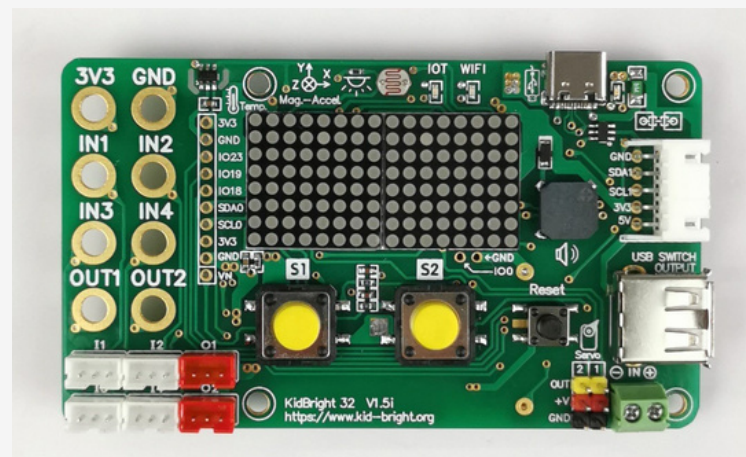
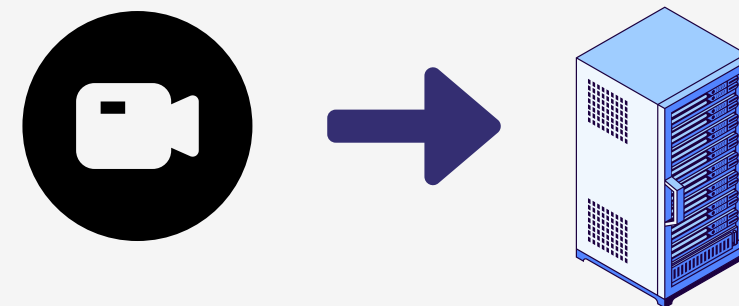


Samsung Galaxy A12

+

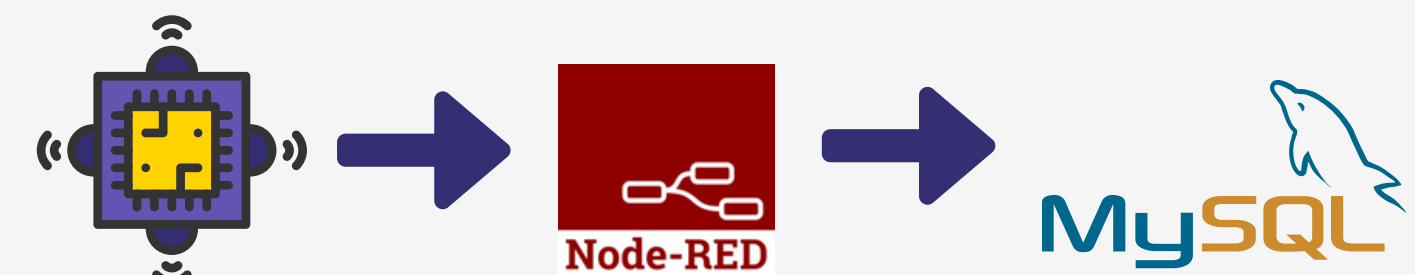
IP Camera Application

1. Send realtime video with RTSP
2. Server Receive video and process the video
3. While processing the video, also save snapshot from video every 10 seconds



Kidbright  
Temperature and Light

1. Collect temperature and light data with built-in sensors
2. Use NODE-RED to collect data along with data from secondary source
3. Save to database





# Data source : Secondary

---

Secondary Source

**API**

Data  
from  
weather.api

**OPENWEATHERMAP.ORG**



## The requested Data

outdoor\_temp

outdoor\_feels\_like

outdoor\_pressure

outdoor\_humidity

outdoor\_weather

outdoor\_description

outdoor\_pm25

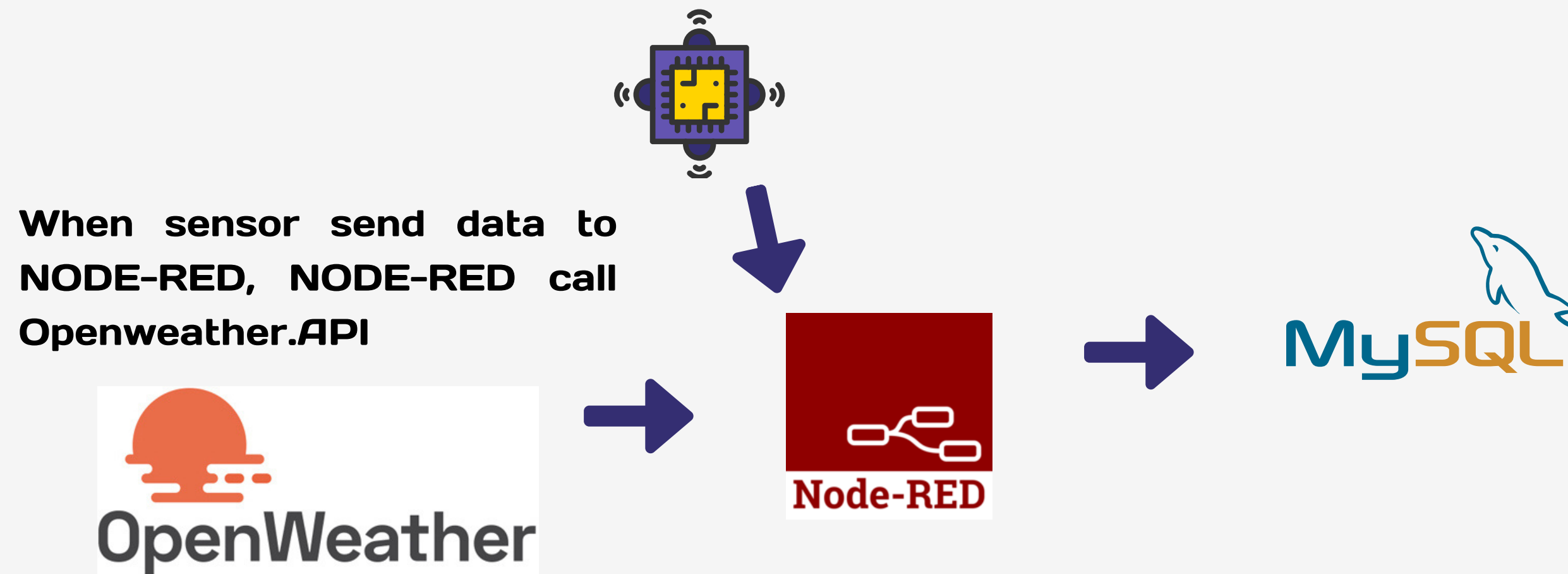
outdoor\_pm10



# Secondary source : Collection Mechanism

---

1. **NODE-RED** wait for sensors to send data
2. **After** receive data from sensor, it fetch data from **Openweather.api**
3. **Save** the needed data in **SQL** database





The background features a dark blue field with intricate magenta geometric patterns. These patterns consist of numerous thin, parallel lines that form a series of nested, slightly offset rectangular shapes, creating a sense of depth and movement. The lines are most prominent on the right side of the image, where they appear to converge towards the top right corner.

# Database Schema



# Database Schema

## Database Scheme

Data	
id	int
timestamp	timestamp
outdoor_temp	float
outdoor_feels_like	float
outdoor_pressure	int
outdoor_humidity	int
outdoor_weather	text
outdoor_description	text
outdoor_pm25	int
outdoor_pm10	int
indoor_temp	float
indoor_light	int

Action	
id	PK
timestamp	timestamp
action	varchar(30)

Prediction	
id	int
timestamp	timestamp
outdoor_temp	float
outdoor_feels_like	float
outdoor_pressure	int
outdoor_humidity	int
indoor_temp	float

## Have 3 tables

- **Data**
- **Action**
- **Prediction**

## Data Table

- **Store data from NODE-RED**

## Prediction Table

- **Use some data from “Data Table” to predict “indoor temperature”**

## Action Table

- **Store action from action model (evaluate action from real-time video)**



The background features a dark blue field with intricate, glowing magenta geometric patterns. These patterns consist of numerous thin, parallel lines that form a series of nested, elongated shapes, creating a sense of depth and movement. The lines are most concentrated in the upper right and lower right corners, where they form a complex, almost crystalline structure. The overall effect is a modern, high-tech aesthetic.

# ***API Overview***



# API Description

---

All API provide with FastAPI, almost all API is REST and one is Websocket



{ REST : API }



Root path of API is /api/v1

API has 4 big router

- camera
  - **CRUD** functions to manage the camera and to observe the camera video.
- weather
  - **Provide** statistics (average, sum, etc.) about each environmental-related data
- recommend
  - **Send** warnings or recommend base on weather data
- action
  - **Send** data about action that we evaluate from camera

# Camera Path

---

**POST**

**/camera/add** Add Camera

**GET**

**/camera/list** List Cameras

**GET**

**/camera/stream/{camera\_id}** Stream Video

**GET**

**/camera/stream/action/{camera\_id}** Stream Action Video

**DELETE**

**/camera/remove/{camera\_id}** Disconnect Camera

**GET**

**/camera/snapshot/{interval}** Get Snapshot



# Weather Path

---

GET

**/weather/** Get Latest Weather Data

GET

**/weather/{days}** Get Weather Data Last N Days

GET

**/weather/indoor/{days}** Get Indoor Data Last N Days

GET

**/weather/outdoor/{days}** Get Outdoor Data Last N Days

GET

**/weather/average/outdoor/{days}** Get Average Outdoor Data

GET

**/weather/average/indoor/{days}** Get Average Indoor Data

GET

**/weather/indoor/predict/** Get Tomorrow Indoor Temp



# Action and Recommend Path

---

**POST**

`/recommend/recommendation/` Get Health Recommendation

**GET**

`/action/week` Get Week Action

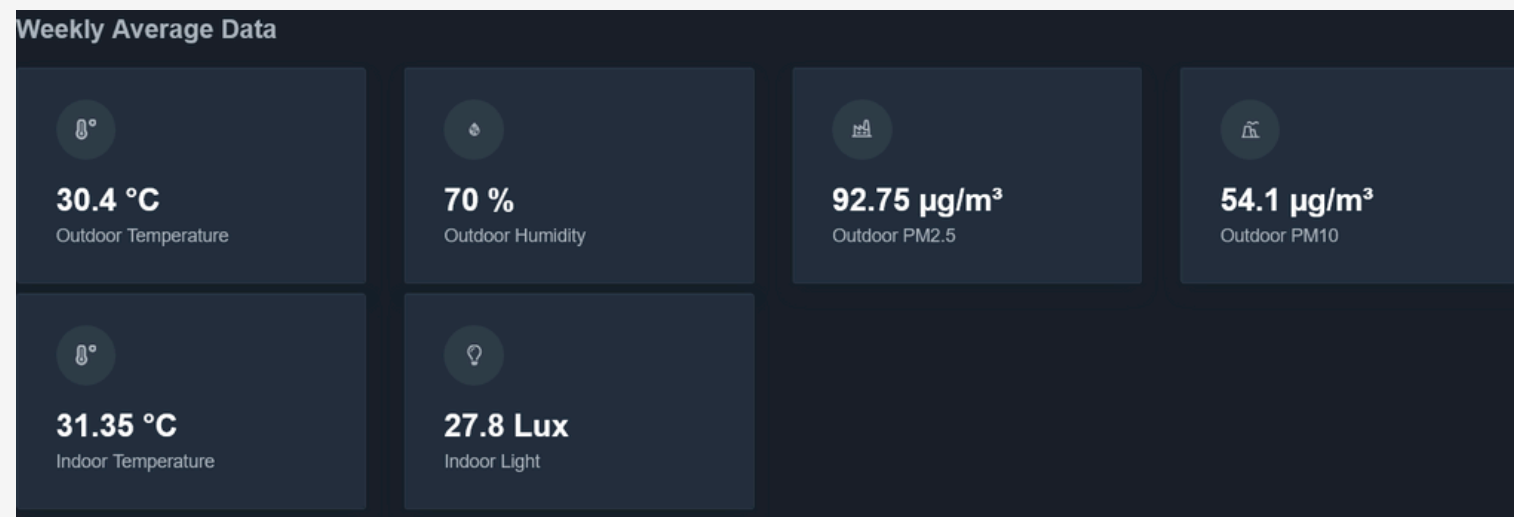
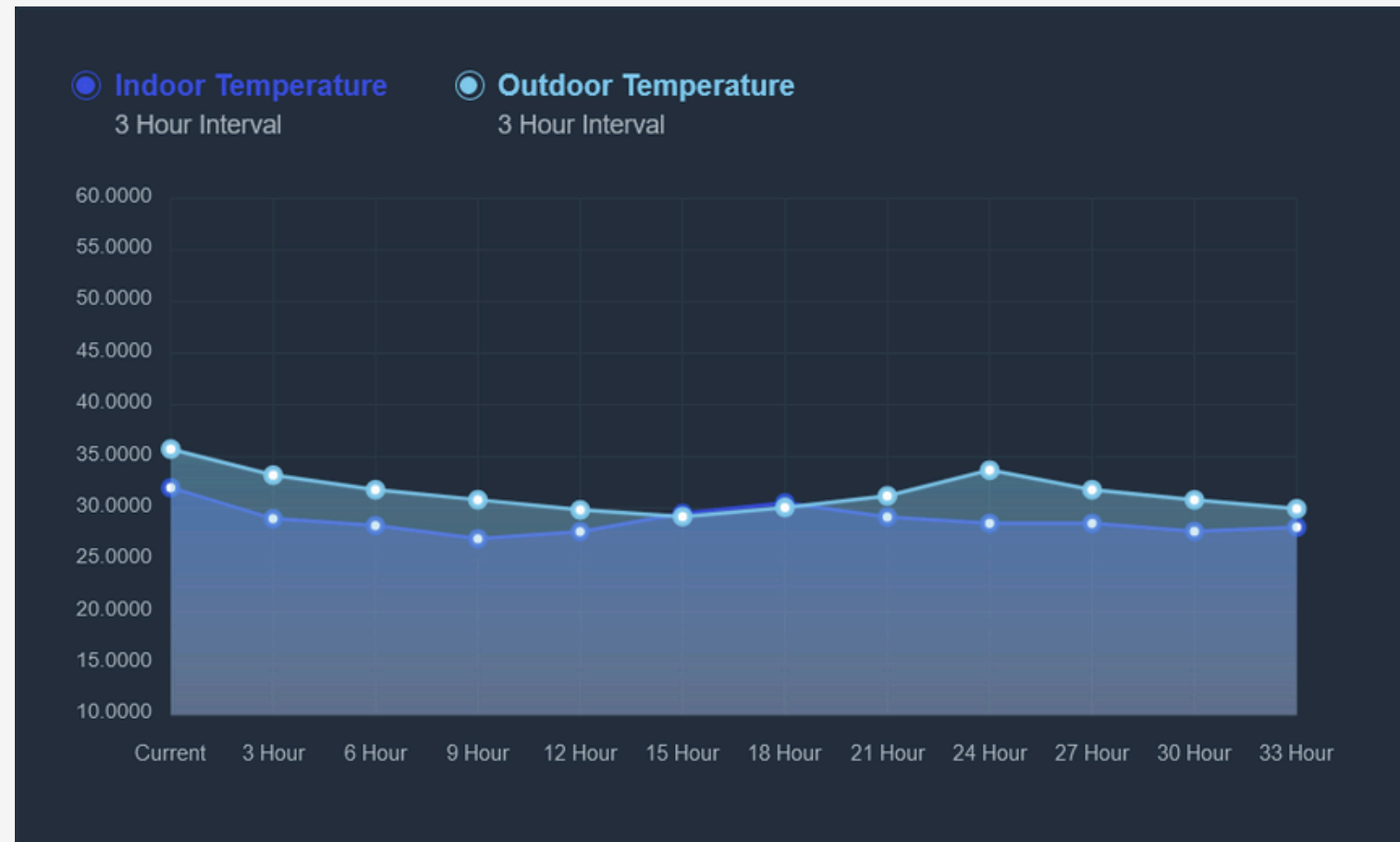


The background features a dark blue field with intricate, glowing magenta geometric patterns. These patterns consist of numerous thin, parallel lines that form a series of nested, slightly offset rectangular frames, creating a sense of depth and movement. The lines are most concentrated on the right side of the image, where they form a dense, almost tunnel-like structure that recedes into the distance.

# **Data Visualization**



# Data Visualization



We have 3 key visualization

- Line plot of 36 hours indoor vs outdoor temperature
- Bar plot of pm 2.5 and pm 10
- Pie chart of personal action



The background features a dark blue field with intricate magenta line art. On the right side, there are several sets of concentric, slightly offset rectangular lines that create a sense of depth and perspective, resembling a stylized architectural structure or a series of nested frames. On the left side, there is a more complex pattern of intersecting lines that form a grid-like structure, possibly representing a wireframe or a digital mesh. The overall aesthetic is modern and technological.

# Demo

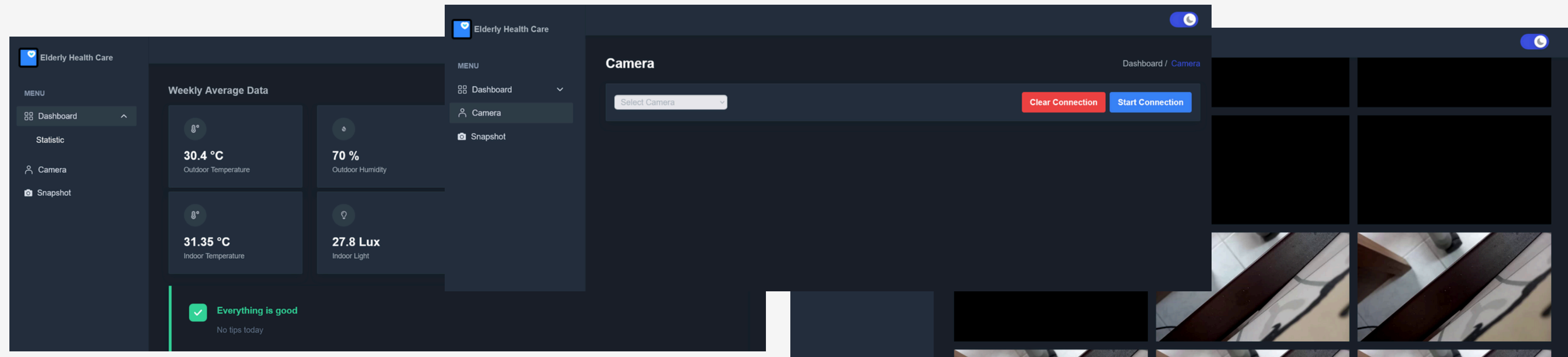


# Key Features

---

**Evaluate action  
of person in  
camera.**

**Provide statistic  
and visualization  
for user**







**THANK YOU**