# **ALERT NOTIFICATION SYSTEM**

**Features**:

* InfluxDB sensor data integration
* OpenHermes/Ollama-powered chatbot
* Streamlit front-end
* Multi-channel alert system (Slack, WhatsApp, Gmail, TEAMS, Pushbullet)

**Data Flow Overview**

## **InfluxDB → Python/Streamlit App → Conditional Logic → Trigger Alert → Channels**

## Alert Channels Implemented

## Slack Alerts

#### Setup:

* Go to Slack → Apps → **Incoming Webhooks**
* Add a webhook and choose the channel (e.g., #weather-alerts)
* Copy the generated Webhook URL

**Python Integration:**

**import requests**

**def send\_slack\_alert(message):**

**webhook\_url = "https://hooks.slack.com/services/XXX/XXX/XXX"**

**payload = {"text": f"⚠️ Weather Alert: {message}"}**

**response = requests.post(webhook\_url, json=payload)**

**return response.status\_code == 200**

## **WhatsApp Alerts**

#### Setup:

* Sign up at [Twilio](https://www.twilio.com/)
* Activate **WhatsApp sandbox**
* Note down:
  + Account SID
  + Auth Token
  + Twilio sandbox number (+14155238886)
  + Verify your recipient number

#### Install:

## **pip install twilio**

## Python Integration:

## **from twilio.rest import Client**

## **def send\_whatsapp\_alert(message):**

## **account\_sid = "YOUR\_SID"**

## **auth\_token = "YOUR\_AUTH\_TOKEN"**

## **client = Client(account\_sid, auth\_token)**

## **client.messages.create(**

## **from\_='whatsapp:+14155238886',**

## **body=f"⚠️ Weather Alert: {message}",**

## **to='whatsapp:+91XXXXXXXXXX'**

## **)**

## **Gmail Alerts**

#### Setup:

* Go to Google Account → Security
* Turn on **2-Step Verification**
* Create an **App Password**
* Use it as your SMTP password

#### Python Integration:

## **import smtplib**

## **from email.mime.text import MIMEText**

## **def send\_gmail\_alert(subject, body):**

## **sender = "your-email@gmail.com"**

## **recipient = "receiver-email@gmail.com"**

## **password = "your-app-password"**

## **msg = MIMEText(body)**

## **msg['Subject'] = subject**

## **msg['From'] = sender**

## **msg['To'] = recipient**

## **server = smtplib.SMTP("smtp.gmail.com", 587)**

## **server.starttls()**

## **server.login(sender, password)**

## **server.send\_message(msg)**

## **server.quit()**

## **Pushbullet**

#### Setup:

* Create account at [Pushbullet](https://www.pushbullet.com/)
* Go to [Settings](https://www.pushbullet.com/#settings) → Generate Access Token
* Install Pushbullet app on phone

#### Install:

## **pip install requests**

## Python Integration:

## **import requests**

## **def send\_pushbullet\_alert(message):**

## **token = "YOUR\_ACCESS\_TOKEN"**

## **headers = {'Access-Token': token, 'Content-Type': 'application/json'}**

## **data = {"type": "note", "title": "⚠️ Weather Alert", "body": message}**

## **response = requests.post("https://api.pushbullet.com/v2/pushes", json=data, headers=headers)**

## **return response.status\_code == 200**

## **Microsoft Teams**

#### Setup:

* Open **Microsoft Teams**.
* Go to the **channel** where you want to post alerts.
* Click on the **⋯ (three dots)** next to the channel name → Choose **Connectors**.
* In the list, find **Incoming Webhook** → Click **Configure**.
* Give it a name (e.g., Weather Alerts) and optionally upload an image.
* Click **Create**.
* **Copy the webhook URL** provided (you’ll use it in your Python script).

## Python Integration:

import requests

def send\_teams\_alert(message):

webhook\_url = "https://outlook.office.com/webhook/your-long-url-here" # Replace with your webhook URL

payload = {

"@type": "MessageCard",

"@context": "https://schema.org/extensions",

"summary": "Weather Alert",

"themeColor": "EA4300",

"title": "⚠️ Weather Alert",

"text": message

}

headers = {'Content-Type': 'application/json'}

response = requests.post(webhook\_url, json=payload, headers=headers)

if response.status\_code == 200:

print("✅ Teams alert sent.")

return True

else:

print(f"❌ Failed to send Teams alert: {response.status\_code}, {response.text}")

return False

**Usage Example:**

if temp\_values and max(temp\_values) > 30:

alert\_msg = f"🔥 High temperature detected: {max(temp\_values):.2f}°C"

send\_teams\_alert(alert\_msg)























