Team Details:

Names	USN
Monisha J	**1MS23CS117**
Risshab Srinivas Ramesh	**1MS23CS152**
Roshani T S Udupa	**1MS23CS155**
Saikiran Krishnamurthy Shet	**1MS23CS159**

RaspberryPi Weather Station

Introduction

The Raspberry Pi Weather Station is an Internet of Things (IoT) project designed to monitor environmental parameters such as temperature, humidity, and atmospheric pressure in real-time. By leveraging the Raspberry Pi's computational capabilities and connectivity, the system collects data from sensors, displays it locally, and transmits it to the ThingSpeak cloud platform for remote monitoring and analysis.

Hardware Components:

1. **Raspberry Pi 3 Model B**: A Raspberry Pi serves as the central computing unit, running Raspberry Pi OS.
2. **SenseHAT**: An add-on board for the Raspberry Pi with built-in environmental sensors.
    a. Temperature Sensor
    b. Humidity Sensor
    c. Pressure Sensor
3. **Power Supply and microSD Card**: For running the Raspberry Pi OS and code.

Software Setup

1. **Raspberry Pi OS**: A Debian-based operating system flashed onto the microSD card. The OS is configured to enable I2C and GPIO interfaces.

2. **ThingSpeak Configuration**: An account is created on ThingSpeak.com, and a new channel is set up with three fields: Field 1 (Humidity), Field 2 (Temperature), and Field 3 (Pressure). The Write API key is obtained from the "API Keys" tab and used in the Python script to upload data.

3. **Python Libraries**:
    a. Adafruit_DHT: For interfacing with the DHT11/DHT22 sensor to read temperature and humidity data.
4. **MATLAB (via ThingSpeak Apps)**: Used for applying regression models (linear/polynomial), smoothing, anomaly detection, or forecasting using scripts.

Workflow

1. Sense HAT sensors read real-time environmental parameters.
    a. Sensors continuously monitor environmental changes in real-time, providing up-to-the-minute readings.
    b. The data is digital and communicated over the I^2C protocol to the Raspberry Pi.
2. Raspberry Pi reads sensor values using a Python script.
    a. Python script is set to run in a loop with time delays for periodic sampling.
3. Collected data is formatted and sent to ThingSpeak via API.
    a. Channel Setup
        i. Create a ThingSpeak Channel with:
            - Field 1: Temperature
            - Field 2: Humidity
            - Field 3: Pressure
        ii. Copy the Write API Key to be used in the Python script.
4. ThingSpeak stores and displays data in real-time graphs.
    a. Data Storage & Visualization
        i. Real-time graphs for each field.
        ii. Customize widgets like gauges, bar charts, or line plots.
        iii. Auto-refresh dashboards every minute for live updates.
5. MATLAB analysis is performed on the ThingSpeak channel to generate:
    a. Histograms (for temperature and humidity)
    b. Linear regression (temperature vs. time)
    c. Polynomial regression (temperature trend estimation)
6. Post-Collection Data Analysis Using Python
    a. Export CSV from ThingSpeak containing:
        i. Timestamp
        ii. Temperature (°C)
        iii. Humidity (%)
        iv. Pressure (hPa)
    b. Read and process the CSV using Python libraries:
        i. pandas for data manipulation
        ii. matplotlib and seaborn for visualization
        iii. scikit-learn and numpy for regression analysis