

# Detailed Development Plan: IoT-Based Air Pollution Monitoring System Using LoRa

---

## Phase 1: Development Environment Setup

Objective: Prepare the hardware and software environment for development.

Tasks:

- Install Arduino IDE and required libraries (`LoRa.h`, `DHT.h`, etc.)
- Install Python 3, pip, Django, and required packages on Raspberry Pi.
- Enable UART and SPI on Raspberry Pi via `raspi-config`.
- Set up Git repository for version control and collaboration.

## Phase 2: Transmitter Node (Arduino UNO)

Objective: Collect sensor data and transmit via LoRa.

Tasks:

- Integrate the MQ135 sensor for gas detection using analog input.
- Integrate the DHT11 sensor for temperature and humidity.
- Interface the SDS011 sensor via UART for PM2.5 and PM10 readings.
- Format data as a comma-separated string for transmission.
- Transmit data using the LoRa RA-01 module via SPI.
- Ensure proper power supply through a buck converter and LDO regulator.
- Implement basic error checking and timing delays.

## Phase 3: Receiver Node (Raspberry Pi)

Objective: Receive data from transmitter and store it.

Tasks:

- Connect LoRa RA-02 to Raspberry Pi via UART or SPI.
- Use Python and pyserial to receive and decode LoRa packets.
- Create a local SQLite or PostgreSQL database.
- Design and implement a `readings` table with appropriate fields.

- Insert received data into the database in real-time.
- Add CRC or simple checksum validation for data integrity.

## Phase 4: Web Application Interface (Django)

Objective: Display real-time and historical data via a web dashboard.

Tasks:

- Create a Django project and app (`airmonitor`, `dashboard`).
- Define models for air quality data.
- Use Django ORM to manage database queries.
- Build views and templates to show live and historical data.
- Integrate Chart.js or Plotly for graphical visualization.
- Enable API endpoints if external access is needed.

## Phase 5: Testing & Optimization

Objective: Ensure system stability and optimize performance.

Tasks:

- Validate sensor accuracy and consistency in test environment.
- Test LoRa signal strength and optimize spreading factor and bandwidth.
- Confirm power efficiency using battery-operated trials.
- Debug PCB connections and re-solder faulty joints.
- Optimize sleep cycles and reduce power usage.
- Conduct full integration tests with transmitter, receiver, and web dashboard.

## Phase 6: Future Enhancements (Optional)

Objective: Plan for long-term scalability and features.

Tasks:

- Add SMS/Email alert system for AQI threshold breaches.
- Integrate cloud-based storage and visualization (e.g., Firebase).
- Develop a mobile application for remote access.
- Implement more accurate AQI algorithms using calibrated sensors.