**AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY (AUST) 141 & 142, Love Road, Tejgaon Industrial Area, Dhaka-1208.**



Department of Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Course No: 4264

Course Title: Internet of Things (IOT) Lab

# Project Workflow and Architecture

Date of Submission: 2/1/2025

## Submitted by,

Group: 2

Md. Tahiadur Rahman Id: 20200204003

Adiba Amin Id: 20200204012

Samia Habib Id: 20200204044

MD. Rafiu Alam Rafi Id: 20200204051

**Project: Smart System for Air Pollution Monitoring with Safe Data Transmission.**

**Workflow for the Smart System:**

1. **Device Enrollment and Authentication**:
   * Register IoT devices like ESP8266/Raspberry Pi and sensors (e.g., MQ135, SDS011, DHT22).
   * Authenticate devices to ensure secure access.
2. **Data Acquisition**:
   * Sensors collect real-time data on air pollutants (e.g., CO₂, PM2.5, PM10) and environmental factors like temperature and humidity.
3. **Data Communication**:
   * Transmit sensor data to the cloud using protocols like MQTT or HTTP over Wi-Fi.
4. **Data Processing**:
   * Process data locally on Raspberry Pi (edge computing) for preliminary analysis or filtering.
   * Securely send processed data to the cloud for advanced analysis.
5. **Cloud Integration**:
   * Store and analyze the data in cloud platforms (e.g., AWS, Azure, ThingsBoard).
6. **Data Visualization**:
   * Display data insights on user-friendly dashboards with charts, alerts, and metrics.
7. **Action and Notification System**:
   * Generate alerts (email/SMS) when pollution levels exceed safe thresholds.
   * Provide actionable insights to stakeholders for preventive measures.

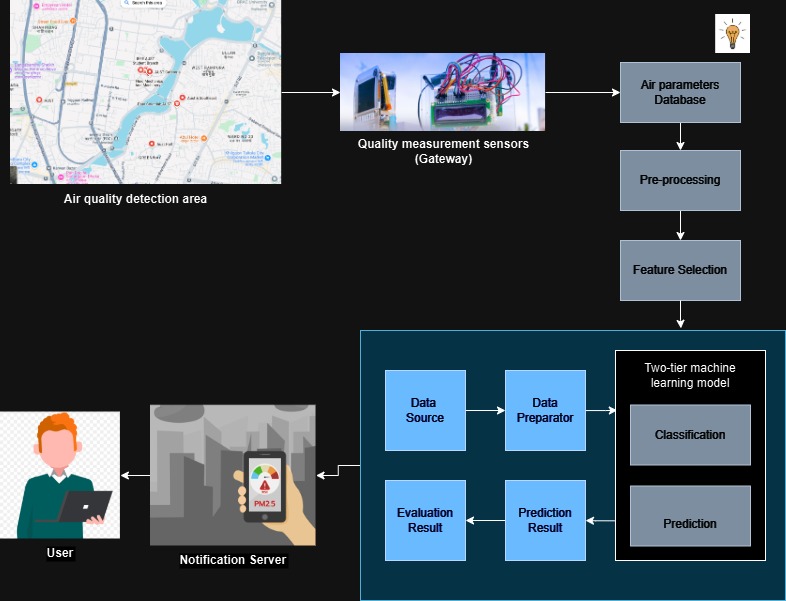


Figure: Workflow

**Architecture for the Smart System:**

1. **Sensing Layer**:
   * **Hardware**: Sensors (MQ135, SDS011, DHT22) to monitor pollutants and environmental conditions.
   * **Microcontrollers**: ESP8266 or Raspberry Pi to interface with sensors.
2. **Network Layer**:
   * Communication protocols like MQTT or HTTP.
   * Transmission via Wi-Fi or cellular networks.
3. **Data-Processing Layer**:
   * Local processing on Raspberry Pi for real-time alerts.
   * Cloud storage and analysis for long-term trends and reporting.
4. **Application Layer**:
   * User-friendly dashboards accessible via mobile or web interfaces.
   * Integration with alert/notification systems.

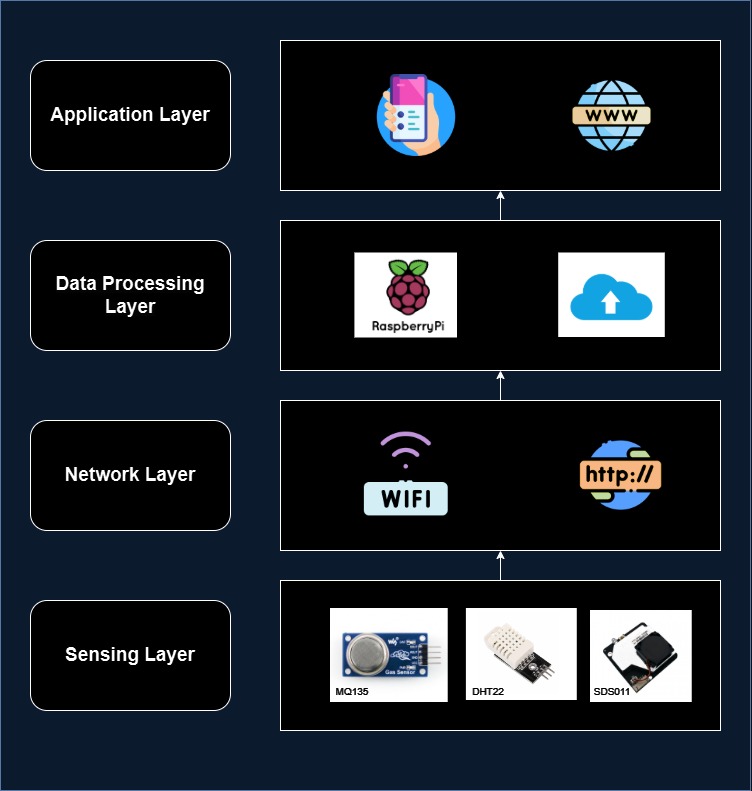


Figure: Building Blocks of IoT Architecture

**Implementation Steps for the Smart System:**

1. **Hardware Setup**:
   * Connect sensors (MQ135, SDS011, DHT22) to the microcontroller.
   * Configure the power supply and communication interfaces.
2. **Microcontroller Configuration**:
   * Install Arduino IDE for coding ESP8266 or Raspberry Pi OS for the Raspberry Pi.
   * Write and deploy firmware to capture sensor readings and transmit them securely.
3. **Communication Protocol Setup**:
   * Configure MQTT/HTTP for data transmission.
   * Secure communication using encryption (AES or TLS).
4. **Cloud Platform Integration**:
   * Set up a ThingsBoard instance or another cloud platform.
   * Configure device profiles and data ingestion pipelines.
5. **Dashboard Development**:
   * Design dashboards with widgets like charts, gauges, and maps.
   * Include real-time updates and historical trends.
6. **Alert System Implementation**:
   * Define thresholds for pollution levels.
   * Configure alerts via SMS, email, or push notifications.
7. **Testing and Deployment**:
   * Test the system with sample data to ensure functionality.
   * Deploy in target areas for real-world data collection and analysis.
8. **Analysis and Modeling**:
   * Incorporate Python models (e.g., Linear Regression, LSTM) to predict vulnerabilities and analyze air quality trends.
   * Evaluate models using metrics like MSE, R-squared, and AQI calculations.