



# KGiSL Institute Of Technology

## NAAN MUDHALVAN

### ***Problem Definition:***

Set up IoT devices to measure air quality parameters like pollution levels and particulate matter. This data can be made publicly available to raise awareness about air quality and its impact on public health. Use any air quality sensor (eg: VOC sensor) to monitor air pollution levels, particulate matters and various other parameters that define air quality. Connect these sensors to a processor such as Arduino or raspberry pi. Connect an esp8266 module to upload this data to cloud. Make this data publicly available. Create a blog to display this data along with visualizations such as graphs.

### **1. Define Project Objectives and Requirements:**

- Clearly define the project's goals and what specific air quality parameters you want to measure (e.g., PM2.5, PM10, VOC levels, temperature, humidity, etc.).
- Determine the geographical locations where you want to deploy these sensors.

### **2. Select the Hardware:**

- Choose the appropriate air quality sensors (e.g., MQ series for VOC, SDS011 for PM2.5 and PM10) based on your project requirements.
- Select a microcontroller platform such as Arduino or Raspberry Pi to interface with the sensors.
- Add an ESP8266 Wi-Fi module to enable data transmission to the cloud.

### **3. Assemble the Hardware:**

- Connect the air quality sensors to the microcontroller following their datasheets and guidelines.

- Set up the ESP8266 module to connect to a Wi-Fi network.

#### **4. Program the Microcontroller:**

- Write the firmware code for the microcontroller to read data from the sensors periodically.
- Use libraries or APIs provided by sensor manufacturers to obtain accurate readings.
- Format the data for transmission to the cloud.

#### **5. Set Up Cloud Storage and Database:**

- Choose a cloud platform to store the collected data (e.g., AWS, Google Cloud, Microsoft Azure).
- Create a database to store the air quality measurements.
- Set up necessary security measures and access controls to protect the data.

#### **6. Implement Data Transmission:**

- Use the ESP8266 module to send the air quality data to the cloud server over a secure connection (e.g., HTTPS or MQTT).
- Implement error handling and data retry mechanisms to ensure data integrity.

#### **7. Data Visualization and Analytics:**

- Develop a script or application to analyze the data and generate visualizations (e.g., graphs, charts, maps).
- Use tools like Python, R, or web development frameworks to create these visualizations.

#### **8. Create a Blog or Website:**

- Set up a blog or website where you can display the air quality data and visualizations.
- You can use platforms like WordPress, Wix, or build a custom website using HTML/CSS and JavaScript.

#### **9. Public Data Access:**

- Make the air quality data publicly accessible through your website.
- Provide user-friendly interfaces and options for users to explore and download the data.
- Consider implementing APIs for data access by developers and researchers.

#### **10. Promote and Maintain:**

- Promote your project to raise awareness about air quality and its impact on public health.
- Maintain the hardware and software components to ensure data accuracy and reliability.
- Consider involving the community and local authorities in the project for a broader impact.

#### **11. Continuous Monitoring and Alerts (Optional):**

- Implement real-time monitoring and alerting mechanisms if you want to notify users about sudden air quality changes or exceedances of pollution thresholds.