1. **Sensor Integration and Calibration:**

Connect the air quality sensors to the

microcontroller pins. Double-check the wiring to avoid loose connections.Implement sensor calibration routines in your code. This might involve taking multiple readings and adjusting the sensor outputs based on calibration curves provided by the sensor manufacturer.

1. **Microcontroller Programming:**

Write the firmware for your microcontroller. Use appropriate libraries for sensor interfacing.Implement error handling to deal with sensor malfunctions or communication issues.Consider implementing a data buffering mechanism in case of temporary loss of internet connectivity.

3.**Internet Connectivity:**

Integrate Wi-Fi or GSM modules with your microcontroller. Configure the modules to connect to your Wi-Fi network or cellular network.Implement secure communication protocols (such as HTTPS) to ensure data privacy and integrity during transmission.

4. **Cloud Integration:**

Set up a cloud server using your preferred cloud service provider.Create API endpoints or MQTT topics for receiving sensor data.Implement server-side scripts (using languages like Python, Node.js, or Java) to handle incoming data, calculate AQI, and store data in a database (such as MySQL or MongoDB).

5. **User Interface Development:**

Develop the user interface using web technologies (HTML, CSS, JavaScript) or mobile app development frameworks (React Native, Flutter).Implement real-time data