# PROJECT TITLE

|  |
| --- |
| Air quality monitoring system |

# STUDENT/TEAM INFORMATION

|  |  |
| --- | --- |
| **Team Name if any:**  **Team # on Canvas you have self-signed-up for:** | **The Outsiders** |
| **Team member 1 (Team Lead)**  (Lastname, Firstname; SDSU email; picture):  Luming Qi  Lqi5593@sdsu.edu | Hall, Scott – scotthall@sdsu.edu |
| **Team member 2**  (Lastname, Firstname; SDSU email; picture): | Nash, Kevin – kevinnash@sdsu.edu |

# 

# ABSTRACT (15 points)

*(Summarize your project (motivation, goals, system design and results). Max 300 words).*

|  |
| --- |
| This project presents an IoT-based air quality monitoring system using a Flask web server hosted on a local device. The system collects environmental sensor data—specifically PM2.5, eCO2, and TVOC—via HTTP POST requests. Incoming data is timestamped and stored in memory, with real-time email alerts triggered when readings exceed defined thresholds. The system also provides a dashboard via an HTML frontend, displaying the latest 20 records for end-user visualization. Flask handles all server-side logic, with endpoints supporting both data ingestion and retrieval. Gmail SMTP is used to send alert notifications when pollutant levels may affect health. This system is lightweight, self-contained, and deployable on devices such as Raspberry Pi. Key contributions include the integration of real-time alerting, modular Flask route design, and basic security features for data submission. |

# INTRODUCTION (15 pts)

## Motivation/Background (3 pts)

*(Describe the problem you want to solve and why it is important. Max 300 words).*

|  |
| --- |
| Indoor and outdoor air pollution poses significant health risks. For environments like classrooms, labs, or shared housing, real-time awareness of air quality is essential. However, many existing commercial monitoring systems are expensive or rely on cloud platforms with privacy trade-offs. This project aims to provide a lightweight, self-hosted alternative using Python and Flask that can run on any local device. By alerting users to harmful levels of pollutants like PM2.5 or CO₂, the system promotes healthier decision-making (e.g., opening windows or using air purifiers) and can help reduce long-term health issues related to poor air quality. This is especially relevant in areas affected by wildfire smoke, industrial activity, or poor indoor ventilation. |

## Project Goals (6 pts)

(Describe the project general goals. Max 200 words).

|  |
| --- |
| Build a Flask server that can receive sensor data via HTTP POST.  Store and timestamp incoming data for historical reference.  Trigger email alerts when air quality exceeds thresholds.  Provide a dashboard displaying the latest sensor readings.  Ensure local deployment without dependency on external cloud services. |

## Assumptions (3 pts)

(Describe the assumptions (if any) you are making to solve the problem. Max 180 words).

|  |
| --- |
| I assume sensor data is sent from a trusted device and the network is secure.  I assume email SMTP access is available through Gmail or equivalent.  I assume the system will monitor air quality in a single room or area.  I assume the number of concurrent users accessing the dashboard is minimal.  I assume no database is needed; in-memory storage suffices for this use case. |

# SYSTEM ARCHITECTURE (20 pts)

*(Describe the final architecture you have implemented listing sensors, communication protocols (Wi-Fi, BLE, ...), cloud services and user interfaces. Include a block diagram of the system. Max 300 words).*

|  |
| --- |
| Write your answer here  Example block diagram: |

# FINAL LIST OF HARDWARE COMPONENTS (5 pts)

*(Write the final list and quantity of the components you have included in your system)*

|  |  |
| --- | --- |
| **Component/part** | **Quantity** |
| SDS011 | 1 |
| SGP30 | 1 |
| DHT22 | 1 |
| LED | 1 |
| Breadboard | 1 |
| ESP32 board | 1 |
|  |  |

# PROJECT IMPLEMENTATION (30 PTS)

## Tasks/Milestones Completed (15 pts)

*(Describe the main tasks that you have completed in this project. Max 250 words).*

|  |  |
| --- | --- |
| **Task Completed** | **Team Member** |
| Wiring, testing each component, platform Io code, flask server code, alerts, webpage | me |

## Challenges/Roadblocks (5 pts)

*(Describe the challenges that you have faced and how you solved them if that is the case. Max 300 words).*

|  |
| --- |
| One challenge we faced was sending email alerts using Gmail. At first, the program couldn’t log in to Gmail's SMTP server. We found out that Gmail blocks apps like this by default, so we had to turn on “App Passwords” in our Gmail settings. After doing that, the emails worked.  Another issue was when the server received bad or incomplete data. This sometimes caused the Flask app to crash. To fix it, we added error handling in the code so that only correct data would be saved and processed.  We also had problems showing the dashboard because the HTML file was in the wrong place. Flask needs HTML files to be inside a folder called templates. Once we moved the file there, the dashboard started working.  Testing the server with curl was confusing at first, especially when using different devices. The server only worked when we used the correct IP address and made sure the port was open. Setting the host to 0.0.0.0 in Flask solved this problem.  Finally, setting up the virtual environment on the remote Ubuntu server and connecting it to VS Code was a bit tricky. But after activating the environment and selecting the right Python interpreter, everything worked fine. |

## 

## Tasks Not Completed (5 pts)

*(Describe the tasks that you originally planned to complete but were not completed. If all tasks were completed, state so. Max 250 words).*

|  |  |
| --- | --- |
| **Task** | **Reason** |
| Phone app, Kauf smart plug | Out of time and struggling |

# WEAK POINTS / FUTURE WORK (15 pts)

*(Mention at least two points of your project that have room for improvement. These points can be additions to the existing project setup (components) or improvement of the current implementation. Max 200 words).*

|  |
| --- |
| Maybe I will complete phone app for this project. |

# SOURCE CODE (25 pts)

*Please include a link to the source code of your project. A link to a repository (like* [*GitHub*](https://www.oit.uci.edu/uci-github/)*) is preferred.*

|  |
| --- |
| https://github.com/Misaka-git/CS596-IOT-FINAL-PROJECT |