**Minor Project**

**Indoor Air Quality Monitoring & Automation System**



**Submitted To:** Prof. Lava Bhargava  
**Submitted By:** Mohd Anas Ansari (ECE, 3rd Year)  
**Department of Electronics & Communication Engineering**  
**MNIT Jaipur**

**Indoor Air Quality Monitoring & Automation System**

**1. Abstract**

Indoor air quality (IAQ) has a direct and measurable effect on health, comfort, and productivity. This project presents a smart, automated system designed to monitor and improve IAQ in real-time using sensors, a microcontroller (ESP32), MQTT protocol, and web technologies. The system senses parameters like temperature, humidity, CO₂ levels, VOCs, and particulates, and then automatically triggers actions (e.g., turning on fans, opening windows) to maintain air quality within healthy thresholds.

**2. Introduction**

With increasing urbanization and closed living environments, indoor air pollution has become a silent but significant health hazard. Traditional air quality monitoring is either manual or semi-automated, lacking real-time responsiveness. Our project aims to solve this with a fully automated system that not only monitors but also reacts to poor air quality levels using intelligent sensing and control.

**3. Problem Statement**

Indoor spaces are affected by several air quality issues:

* **Mold Growth** due to humidity and lack of airflow.
* **Stale Air / CO₂ Buildup** in closed environments.
* **VOC Accumulation** from cleaning agents or chemicals.
* **Spoilage due to Temperature Fluctuations**.
* **Dust and PM2.5/PM10 accumulation**.
* **CO₂ Accumulation** in crowded rooms.  
  Each of these can lead to serious health impacts, such as respiratory issues, allergies, fatigue, or even long-term illness. Hence, a system is required that can detect these problems and automatically initiate corrective actions.

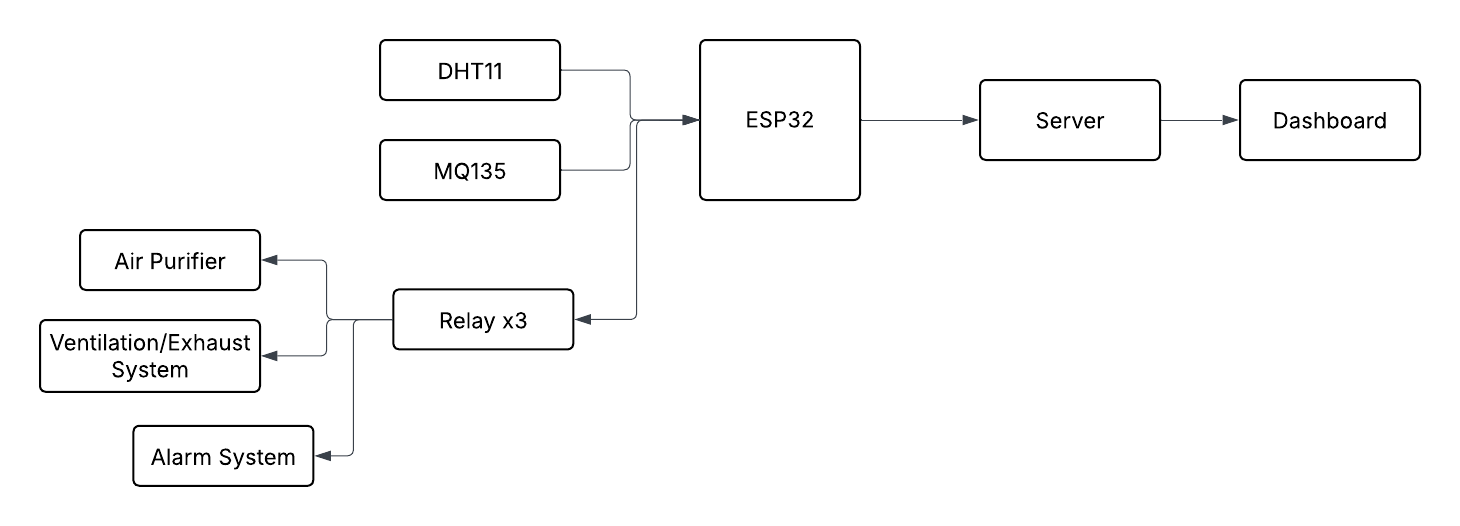
**4. Project Objectives**

* To develop a real-time, responsive system for monitoring indoor air quality.
* To alert users when harmful thresholds are crossed.
* To automatically control ventilation and purification devices based on sensor data.
* To provide a visual dashboard for live data monitoring and system status.

**5. System Overview**

**5.1 Architecture Diagram (Descriptive)**

* **Sensors Layer:** Detects environmental parameters (temperature, humidity, gas, dust, etc.).
* **Microcontroller (ESP32):** Reads sensor data and communicates over WiFi.
* **MQTT Broker (Mosquitto):** Handles publish/subscribe messaging.
* **Actuators (Relays, Fans):** Perform corrective actions based on data.
* **Dashboard:** Displays real-time air quality data using a RESTful API.
* **Feedback Loop:** Continuously monitors and adjusts based on current readings.



**6. Hardware Components**

| **Component** | **Description** |
| --- | --- |
| **ESP32** | WiFi-enabled microcontroller for IoT tasks |
| **DHT11** | Temperature and Humidity Sensor |
| **MQ135** | Air Quality Sensor – detects gases like NH₃, NOₓ, alcohol, benzene, smoke |
| **Relays** | Electromagnetic switches to control AC appliances with microcontroller logic |
|  | Fig 1.1 : ESP-32 DEV MODULE    Fig 1.2 : DHT11 SENSOR    Fig 1.3 : MQ135 SENSOR  https://robotools.in/wp-content/uploads/2022/10/5V-2-Channel-Relay-Module-1-1200x1200.jpg  Fig 1.4 : 5v Dual Channel Relay |

**7. Software Components**

| **Software** | **Function** |
| --- | --- |
| **Arduino IDE** | Programming ESP32 microcontroller |
| **Python** | Handling REST APIs and data processing |
| **Eclipse Mosquitto** | MQTT Broker for communication |
| **HTML/CSS/JS** | Creating a web-based real-time dashboard |

**8. Communication Protocol – MQTT**

**Key Features:**

* **Lightweight and Efficient** for low-power devices like ESP32.
* **Publish/Subscribe Model** allows multiple devices to send and receive data asynchronously.
* **Broker-Based Architecture** ensures central message routing (Mosquitto used).
* **QoS Levels:**
  + QoS 0: At most once
  + QoS 1: At least once
  + QoS 2: Exactly once
* **Low Latency** ensures timely activation of actuators when thresholds are crossed.

**9. Working Principle**

1. Sensors continuously collect environmental data.
2. ESP32 reads sensor values and publishes them to the MQTT broker.
3. A Python backend listens to these MQTT topics and evaluates conditions.
4. If thresholds are breached (e.g., CO₂ > 800 ppm), actuators are triggered.
5. Dashboard fetches latest readings and actuator status via REST API.
6. This process repeats to maintain ideal air quality.

**10. Real-Life Application Scenarios**

| **Problem** | **Sensor Used** | **Automation Action** |
| --- | --- | --- |
| Mold from high humidity | DHT11 | Turns on dehumidifier |
| CO₂ buildup in closed spaces | MQ135 | Activates exhaust fan |
| High VOC concentration | MQ135 | Starts air purifier |
| Spoilage in temperature storage | DHT11 | Turns on air conditioning or heater |
| PM2.5 or dust accumulation | PM Sensor | Engages HEPA air filtration system |

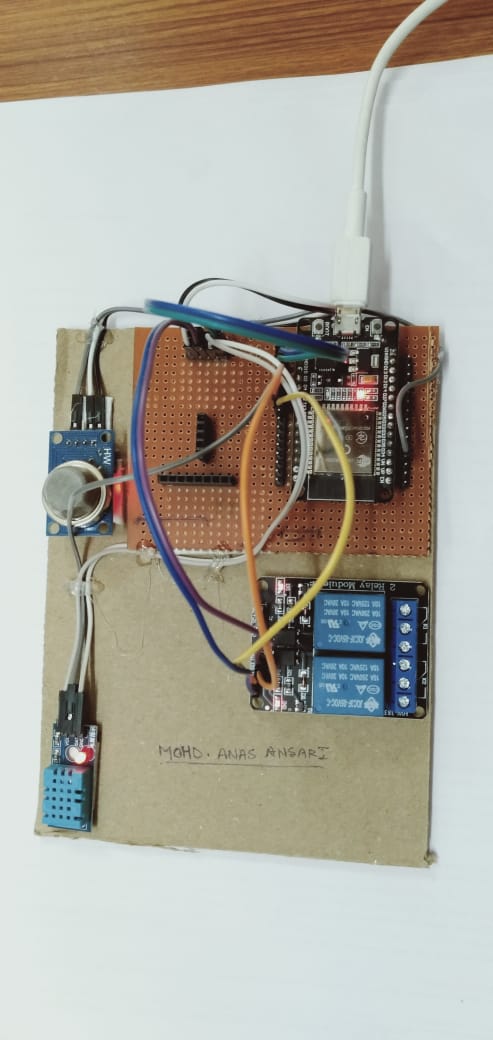
**11. Future Enhancements**

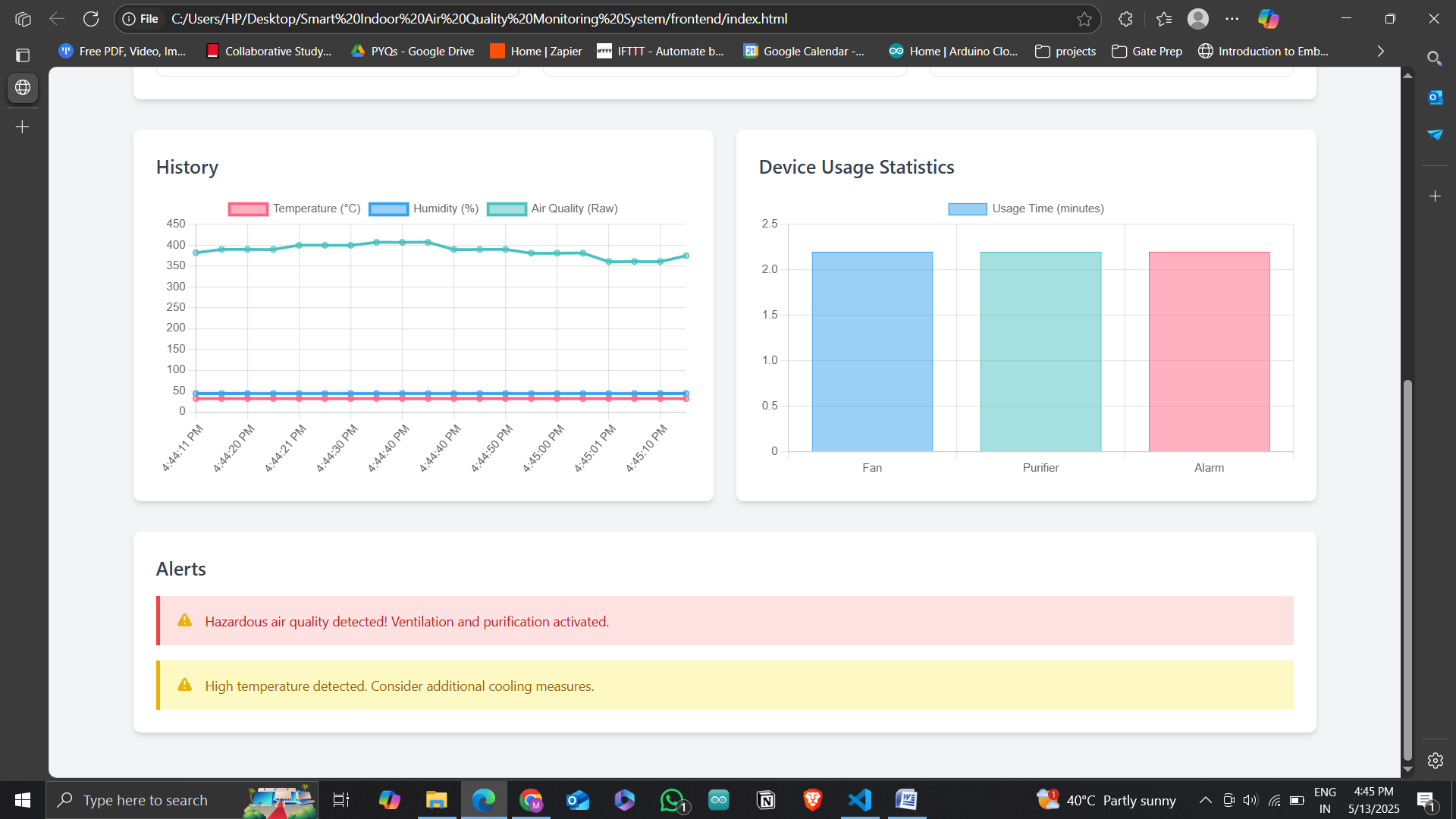
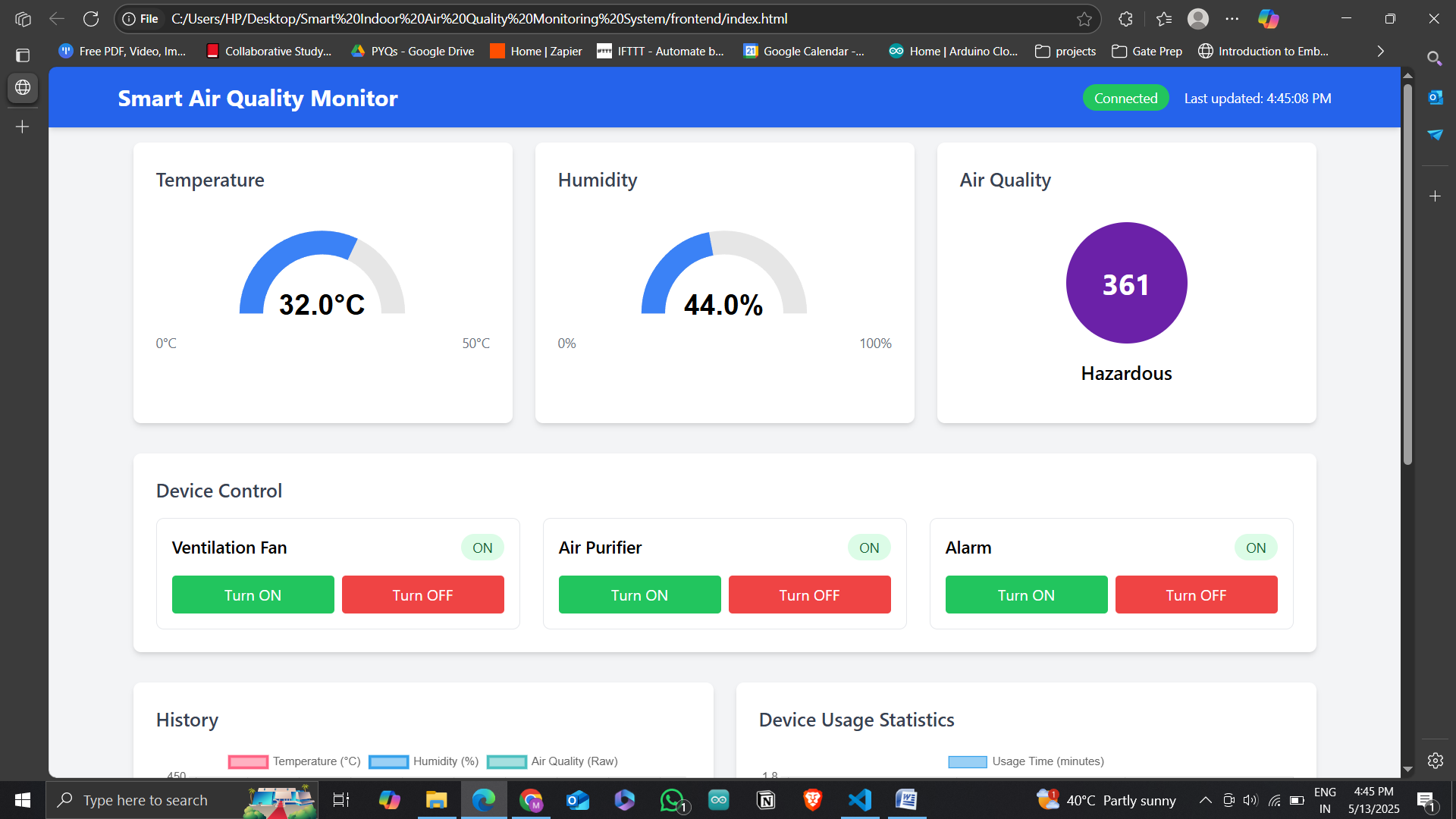
* **Hardware:**
  + Replace DHT11 with DHT22 or BME280 for higher accuracy.
  + Add PM2.5 sensor for fine particulate detection.
* **Software:**
  + Integrate **mobile app** for real-time control.
  + Connect to **cloud-based MQTT brokers** like AWS IoT, Adafruit IO.
  + Use **machine learning** for predictive control (e.g., RNN/LSTM).
  + Add **voice assistants** (Google Assistant, Alexa) for smart home integration.
  + Real-time **alerts/notifications** via SMS, email, or push notifications.

**12. Results & Discussion**

The system accurately monitored indoor air quality and responded effectively to environmental changes.

* **Sensor Readings:**
  + Temperature: 32.0°C
  + Humidity: 44.0%
  + AQI: 361 (*Hazardous*)
* **Automation Response:**  
  On detecting high AQI and temperature, the system automatically turned ON:
  + Ventilation Fan
  + Air Purifier
  + Alarm
* **Usage Stats:**  
  Devices ran for approximately 2.2–2.3 minutes, indicating prompt and effective operation.
* **Graph Trends:**  
  Live graphs showed consistently high AQI and moderate humidity, validating the need for real-time monitoring.
* **Alerts:**  
  Clear warnings were shown for hazardous air and high temperature, enhancing user awareness.

****

****

**13. Conclusion**

This project demonstrates an effective and scalable solution for indoor air quality monitoring using IoT. By combining sensors, microcontrollers, real-time protocols, and automation logic, we can ensure a healthier living and working environment. With future upgrades and cloud integration, this system can be deployed in homes, offices, hospitals, and industrial settings.

**14. References**

1. Indoor Air Quality Monitoring & Automatic Ventilation System
2. Forecasting Energy Consumption using AI
3. Energy Efficient Control Methods For HVAC Systems
4. Indoor Air Quality Monitoring & Controlling System based on IoT & Fuzzy Logic
5. Intelligent Edge: AI & ML in Edge Computing
6. AI-Driven Edge Computing
7. Intelligent HVAC System
8. Review of Energy Efficient HVAC Technologies For Sustainable Buildings