**AIM OF PROJECT –**

To develop an automated Fire Detection System that can detect the presence of flames and smoke in buildings, provide real-time alerts, and visualize data on a cloud-based dashboard for monitoring and safety management.

**PROBLEM STATEMENT –**

Develop a prototype of a Fire Detection System which can –

* Detect files in buildings
* Provide real time detection of flames and
* Transmit alerts and display in dashboards using Cloud

**SCOPE OF THE SOLUTION –**

The scope of the Fire Detection System solution encompasses several key areas, each addressing different aspects of fire safety and monitoring. Here’s a detailed breakdown:

1. **Fire Detection Capabilities:**
   * **Flame Detection:** Use of infrared or ultraviolet flame sensors to detect flames within a specified range.
   * **Smoke Detection:** Implementation of smoke detectors that can identify smoke particles from various types of fires.
   * **Temperature Monitoring:** Integration of temperature sensors to monitor abnormal temperature rises indicative of a fire.
2. **Real-Time Monitoring:**
   * Continuous monitoring of sensor data to provide immediate feedback on fire risk.
   * Implementation of algorithms to analyze data and trigger alerts when thresholds are crossed.
3. **Alert System:**
   * Automated alert notifications sent to designated personnel (e.g., building management, emergency services) through SMS, email, or mobile app notifications.
   * Local alarms, such as buzzers or flashing lights, to alert occupants in the immediate vicinity.
4. **Cloud Integration:**
   * Secure data transmission from sensors to the cloud for storage and analysis.
   * Use of cloud services to facilitate real-time data processing, ensuring quick response to alerts.
5. **Dashboard Development:**
   * Creation of a user-friendly web-based dashboard that displays:
     + Real-time sensor data (flame presence, smoke levels, temperature).
     + Historical data and trends for analysis and reporting.
     + Status of the entire system, including active alerts and sensor health.
6. **Scalability and Flexibility:**
   * Design of the system to easily accommodate additional sensors and monitoring points, allowing for expansion to larger buildings or multiple locations.
   * Configurable thresholds for sensor alerts to cater to different environments (e.g., residential, commercial, industrial).
7. **User Roles and Access Control:**
   * Implementation of different user roles (e.g., admin, user) with varying access levels to the dashboard and alert settings.
   * Ability to customize alert settings based on user preferences.
8. **Data Security:**
   * Ensuring data privacy and security through encryption and secure communication protocols.
   * Compliance with relevant regulations (e.g., GDPR, local fire safety regulations).
9. **Maintenance and Updates:**
   * Provision for regular system maintenance, including updates to software and firmware to enhance functionality and security.
   * Remote diagnostics to troubleshoot and resolve issues without needing physical access to the system.
10. **Integration with Other Safety Systems:**
    * Potential for integration with existing building management systems (BMS) for holistic safety monitoring.
    * Compatibility with other safety devices, such as sprinklers or fire alarms, to trigger coordinated responses.

**REQUIRED COMPONENTS TO DEVELOP SOLUTIONS -**

#### 1. Hardware Components

* **Microcontroller Board**:
  + **Arduino Uno** or **Raspberry Pi 4**: To process sensor data and manage communications.
* **Sensors**:
  + **Flame Sensor**: Infrared (IR) or ultraviolet (UV) sensor for flame detection.
  + **Smoke Detector**: Optical or ionization smoke sensor for detecting smoke.
  + **Temperature Sensor**: (e.g., DHT11 or LM35) to monitor temperature changes.
* **Communication Module**:
  + **Wi-Fi Module**: (e.g., **ESP8266** or **ESP32**) to enable internet connectivity for data transmission.
* **Power Supply**:
  + **Power Adapter** or **Battery Pack**: To supply power to the microcontroller and sensors.
* **Optional Components**:
  + **Buzzer**: For local alarm sounds.
  + **LED Indicators**: To show the status of the system (normal, alert).

#### 2. Software Components

* **Programming Environment**:
  + **Arduino IDE**: For programming Arduino boards.
  + **Python**: If using Raspberry Pi for sensor data processing.
* **Data Processing Libraries**:
  + Libraries for interfacing with sensors (e.g., DHT for temperature and humidity).
* **Dashboard Development**:
  + **HTML/CSS/JavaScript**: For creating a custom web dashboard.
  + **Frameworks**: (e.g., React, Angular) for enhanced web application features.

#### 3. Cloud Environment

* **Cloud Platform**:
  + **AWS IOT**, **Google Cloud IOT**, or **Azure IOT**: For handling data transmission and processing.
* **Database Service**:
  + **Firebase** or **AWS Dynamo DB**: To store alerts and historical sensor data.
* **Data Visualization Tools**:
  + **Grafana** or **Power BI**: For creating dashboards to visualize sensor data and alerts.
* **Alerting Services**:
  + **Twilio** or **AWS Simple Notification Service (SNS)**: For sending SMS/email alerts.

**CODE FOR SIMULATION –**

**#include <ESP8266WiFi.h>**

**#include <WiFiClient.h>**

**#include <AdafruitIO.h>**

**#include <AdafruitIO\_WiFi.h>**

**// Replace these with your network credentials**

**const char\* ssid = "realme8";**

**const char\* password = "lovish@123";**

**// Adafruit IO credentials**

**#define IO\_USERNAME "lovish123"**

**#define IO\_KEY "aio\_xcmf500xG3wZBwIZpYv85Ra5NS3T"**

**// Create an Adafruit IO object**

**AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, ssid, password);**

**#define SMOKE\_SENSOR\_PIN A0 // Analog pin connected to the smoke sensor**

**#define BUZZER\_PIN 4 // GPIO 4 (D2)**

**#define LED\_PIN 5 // GPIO 5 (D1)**

**AdafruitIO\_Feed \*smokeFeed = io.feed("smoke-level"); // Replace with your feed name**

**void setup() {**

**Serial.begin(115200);**

**pinMode(BUZZER\_PIN, OUTPUT);**

**pinMode(LED\_PIN, OUTPUT);**

**// Connect to Adafruit IO**

**Serial.print("Connecting to Adafruit IO...");**

**io.connect();**

**}**

**void loop() {**

**io.run(); // Allow Adafruit IO to process any incoming messages**

**int smokeValue = analogRead(SMOKE\_SENSOR\_PIN); // Read the smoke sensor value**

**Serial.print("Smoke Level: ");**

**Serial.println(smokeValue); // Print the smoke level to the Serial Monitor**

**// Send smoke level to Adafruit IO**

**smokeFeed->save(smokeValue); // Send the value to Adafruit IO**

**// Check if the smoke value exceeds a threshold (e.g., 300)**

**if (smokeValue > 10) {**

**digitalWrite(LED\_PIN, HIGH); // Turn on LED**

**digitalWrite(BUZZER\_PIN, HIGH); // Turn on Buzzer**

**sendAlert(); // Call function to send alert**

**} else {**

**digitalWrite(LED\_PIN, LOW); // Turn off LED**

**digitalWrite(BUZZER\_PIN, LOW); // Turn off Buzzer**

**}**

**delay(500); // Wait for half a second before the next reading**

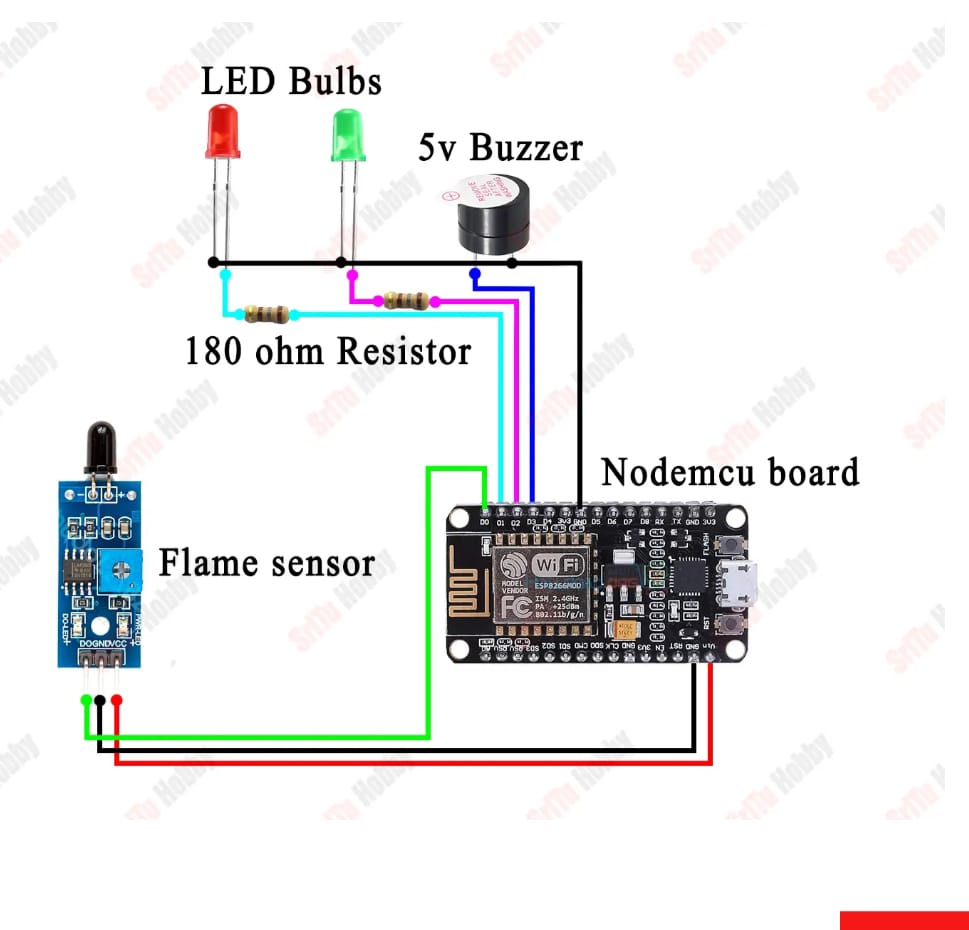
**}**

**void sendAlert() {**

**Serial.println("Alert! Smoke detected!");**

**}**

**SIMULATION IMAGE –**

****

**HARWARE KIT USED –**

1. **Microcontroller Board –**

* **Arduino UNO :** Suitable for simple steps with basic sensors
* **ESP8266 Node MCU Lolin :** (CH340) Ideal for projects requiring built-in Wi-Fi and higher processing capabilities.

1. **Sensors –**

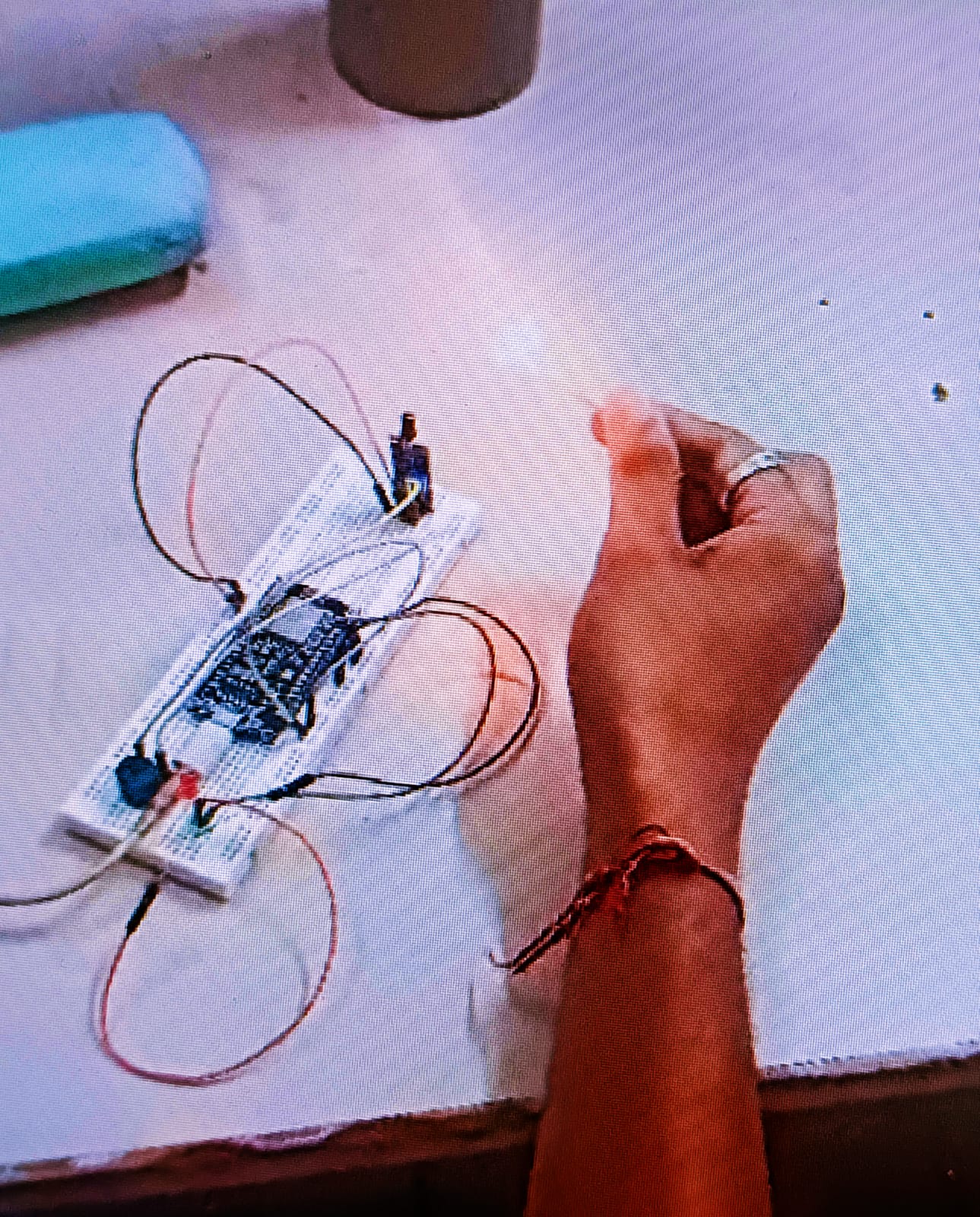
* **Fire/Flame Sensor Module :** (eg KY-026) To detect flames.

1. **Additional Components –**

* **840 Tie Point Breadboard GL12 :** For Prototyping .
* **Jumper Wires :** For Connection .
* **Jumper Cable :** 1M – 1M
* **Jumper Cable :** 1M – 1F
* **180 E Resistor :** As needed for sensor circuits .
* **10 mm Goli Buzzer And LED 5 mm :** For local alerts and indications .

**SCREEN CAPTURES IN DEMO –**

****



**CONCLUSION –**

The intelligent Fire Detection System not only enhances safety and reliability but also aligns with modern smart building trends. By utilizing IOT technologies, the system exemplifies how innovation can significantly improve traditional safety measures. As urban areas become more densely populated and the complexity of building infrastructures increases, the need for robust, efficient fire detection systems becomes ever more critical.

In conclusion, this prototype is a step toward a safer built environment. It empowers users with timely information, reduces risks associated with fire incidents, and fosters a proactive approach to safety management. With ongoing advancements in technology and continued focus on innovation, such systems will play a vital role in protecting lives and property in the future.

* **NOTE –** The video of demo is attached in project portal.

**NAME OF GROUP 7 MEMBERS:**

* **RACHIT SINGLA (2310996782)**
* **LOVISH (2310996786)**
* **MANDEEP(2310996784)**
* **ARMAAN (2310996785)**
* **SHASHANK (2310996783)**

****

****