**Internet of Things Fundamentals**

*Subject Project*

BS AI 6th Smester SP-25 (AIE-3079)

Date:

**Project Title:**

Crowd Monitoring System

**Group Name/no.: Tech Titans**

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| Members | Registration no | Name | Signature |
| **Member-1 (Leader)** | **22\_ntu\_cs\_1334** | **Amish Maqbool** |  |
| **Member-2** | **22\_ntu\_cs\_1357** | **M Sufyan** |  |
| **Member-3** | **22\_ntu\_cs\_1359** | **M Ammad** |  |
| **Member-4** | **22\_ntu\_cs\_1352** | **MahNoor** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Contributions in % of each Team Members for each component | | | | | |
|  | | Member-1 | Member-2 | Member-3 | Member-4 |
| Distribution Components | | Name | Name | Name | Name |
| Coding | ESP32-coding |  |  |  |  |
| Python Coding |  |  |  |  |
| UI Design | |  |  |  |  |
| Database | |  |  |  |  |
| Cloud Integration | |  |  |  |  |
| IoT Gateway | |  |  |  |  |
| Edge Processing | |  |  |  |  |
| Documentation | |  |  |  |  |
| Presentation  Design | |  |  |  |  |
| Replace for other contribution | |  |  |  |  |
| Replace for other contribution | |  |  |  |  |
| Replace for other contribution | |  |  |  |  |
| Replace for other contribution | |  |  |  |  |

*To be filled by the evaluator*

# Team-Based Evaluation (60 Marks)

|  |  |  |
| --- | --- | --- |
| Criteria | Obtained Marks | Out of |
| System Design & Architecture |  | 10 |
| Hardware Integration & Circuit Setup |  | 10 |
| IoT Gateway and Cloud Communication |  | 10 |
| Working Prototype Demonstration |  | 10 |
| Performance & Reliability Testing |  | 10 |
| Presentation |  | 10 |
| Total (Team-Based) |  | 60 |

# Individual-Based Evaluation (40 Marks per Member)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Member 1 | Member 2 | Member 3 | Member 4 |
| Criteria |  |  |  |  |
| Understanding of the Project & Role | /10 | /10 | /10 | /10 |
| Code Contribution and Explanation | /10 | /10 | /10 | /10 |
| Q/A VIVA | /10 | /10 | /10 | /10 |
| Documentation/Reporting & Communication | /10 | /10 | /10 | /10 |
| Total (Individual-Based) | /40 | /40 | /40 | /40 |
| Total Overall (60+40) | /100 | /100 | /100 | /100 |
| Weightage Lab Grade (50) |  |  |  |  |

# Abstract / Executive Summary

This project implements a **Crowd Monitoring System** using an **ESP32 microcontroller**, infrared sensors (or ultrasonic), and environmental sensors to detect **temperature and humidity**. The main goal is to count the number of people in a room or hall in **real-time** and display the live data (people count, temperature, humidity) on a **mobile app**. The app is connected to **Firebase Realtime Database**, which synchronizes the hardware data through a **Python script** running on a local machine or server.  
This system is cost-effective, scalable, and provides reliable crowd data for smart building management, safety, and comfort.

# 2. Table of Contents

1. Abstract / Executive Summary  
2. Table of Contents  
3. Introduction  
4. Literature Review  
5. Methodology / System Design  
6. Implementation  
7. Results & Discussion  
8. Testing & Validation  
9. Conclusion & Future Work  
10. References  
11. Links

# 3. Introduction

**Background & Motivation:**  
Crowd monitoring is critical in smart buildings to optimize energy usage, ensure safety, and manage occupancy efficiently. Manual counting is impractical for real-time monitoring. IoT provides an automated, low-cost solution.

**Problem Statement:**  
Traditional occupancy monitoring solutions are expensive and complex. The goal is to build an affordable system that can monitor people count and environmental conditions in real-time with remote access via a mobile app.

**Project Goals:**

* Detect people entering and exiting a room/hall.
* Measure temperature and humidity.
* Send data to Firebase Realtime Database.
* Display live data on a custom-built mobile app.
* Connect hardware using Python for integration.

# 4. Literature Review (Optional)

* **Relevant IoT Concepts:** Use of ESP32 for Wi-Fi connectivity, IR/Ultrasonic sensors for counting, DHT11/DHT22 for temp/humidity.
* **Similar Projects:**
  + People counter with IR sensors.
  + Smart home occupancy detection.
  + Firebase-based IoT monitoring.

# 5. Methodology / System Design

## 5.1 Hardware Components

**Components Used:**

* ESP32 Dev Board
* IR Sensor Pair (or Ultrasonic)
* DHT11/DHT22 sensor
* Wires, Breadboard
* PC/laptop for Python bridge
* Wi-Fi Network

## 5.2 Software Design

[Start]

↓

[ESP32 Powers Up]

↓

[Read IR Sensors]

↓

[Update People Count]

↓

[Read Temp & Humidity]

↓

[Send Data to Python Script]

↓

[Python Updates Firebase]

↓

[Firebase Updates Mobile App]

↓

[Display on App in Real-Time]

↓

[Repeat]

# 6. Implementation

**Libraries/Tools Used:**

* **ESP32 IDE:** Arduino IDE, PlatformIO
* **Python:** firebase\_admin SDK, pyserial
* **Firebase:** Realtime Database
* **App:** Android Studio, Flutter, or MIT App Inventor

**Pseudocode Example:**

plaintext

CopyEdit

LOOP:

Read IR sensors → Determine entry/exit → Update count

Read DHT11 → Get Temp & Humidity

Send data over Serial to Python script

Python script uploads to Firebase

App reads Firebase & displays

**Step-by-Step Setup:**

1. Connect IR sensors at door/entry.
2. Connect DHT sensor to ESP32.
3. Upload Arduino code to ESP32.
4. Run Python script on PC to read Serial data and push to Firebase.
5. Build and run the mobile app to show live Firebase data.

**Sample Code Snippets:**

cpp

CopyEdit

#include <DHT.h>

DHT dht(4, DHT11);

void setup() {

Serial.begin(115200);

dht.begin();

}

void loop() {

float t = dht.readTemperature();

float h = dht.readHumidity();

// Read IR sensors and update count logic

int peopleCount = ...;

// Send to Python over Serial

Serial.print(peopleCount);

Serial.print(",");

Serial.print(t);

Serial.print(",");

Serial.println(h);

delay(2000);

}

**Python Script (Main Part):**

python

CopyEdit

import serial

import firebase\_admin

from firebase\_admin import credentials, db

cred = credentials.Certificate('path/to/serviceAccountKey.json')

firebase\_admin.initialize\_app(cred, {

'databaseURL': 'https://your-firebase.firebaseio.com/'

})

ser = serial.Serial('COM3', 115200)

while True:

line = ser.readline().decode('utf-8').strip()

count, temp, hum = line.split(',')

ref = db.reference('/')

ref.update({

'people\_count': count,

'temperature': temp,

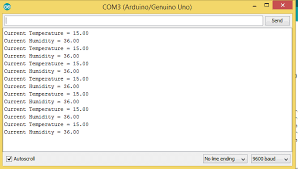
'humidity': hum

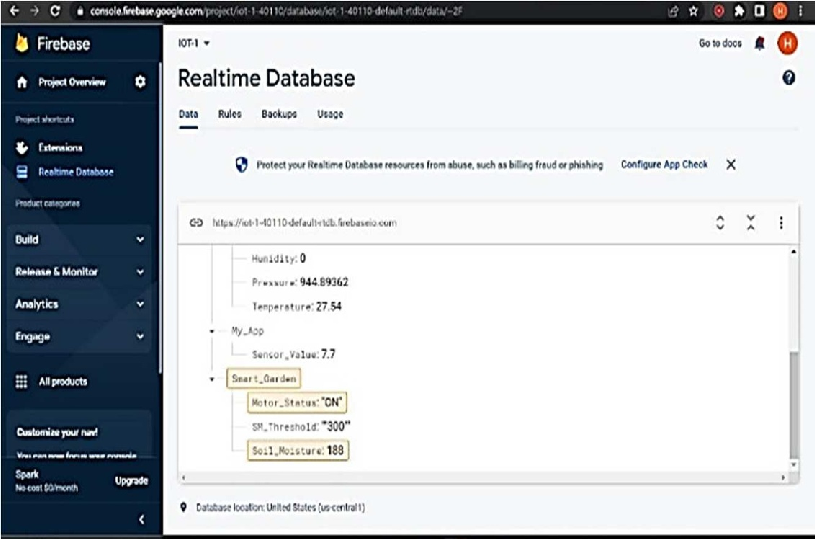
})

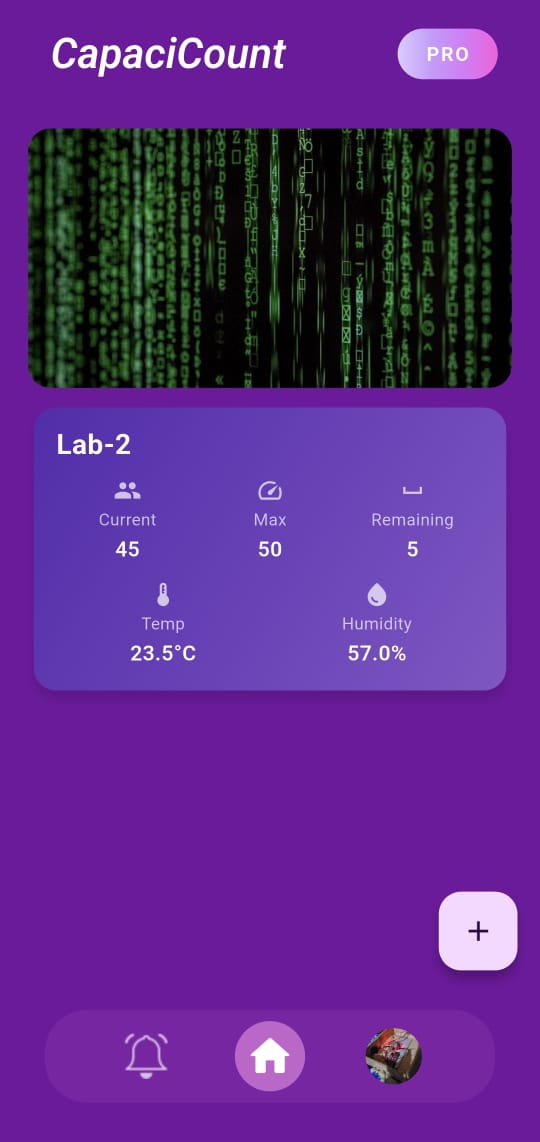
**Challenges & Solutions:**

* *Wi-Fi issues:* Ensured stable connection.
* *False triggers:* Calibrated IR sensors for better accuracy.

# Results & Discussion







# Testing & Validation / Limitations

**Test Cases:**

* Multiple people entry/exit.
* Temperature variations.
* Network disconnection recovery.

**Limitations:**

* IR sensors might double count if entry/exit is too fast.
* Relies on continuous Wi-Fi connection.

# Conclusion & Future Work

**Key Takeaways:**

* Affordable, real-time crowd monitoring.
* Simple integration with Firebase.
* Easy to expand or upgrade.

**Future Work:**

* Add camera + AI for better accuracy.
* Cloud dashboard with analytics.
* Add alerts if room overcrowded.

# References

1. Firebase Docs
2. [ESP32 Arduino Core](https://github.com/espressif/arduino-esp32)
3. YouTube tutorials for ESP32 + Firebase

# 11. Links