# TITLE PAGE

Project Title: HomeSense - Smart Home Environmental Monitoring System

Author: [Student Name]

Department: [University / Course Name]

Institution: [University Name]

Date: [Month, Year]

# 1. INTRODUCTION

Modern households rely on environmental monitoring systems to maintain indoor comfort, health, and safety. HomeSense provides an Internet of Things based multi sensor solution using ESP32 modules distributed across rooms. The system collects temperature, humidity, air quality, and motion data and transmits readings over MQTT to a backend with real time dashboards and alerting.

# 2. SYSTEM OVERVIEW

HomeSense uses low power wireless nodes that continuously sense environmental conditions and publish data to an MQTT broker. A backend stack consisting of a processing service, time series database, and visualization platform enables real time insights and long term storage.

# 3. HARDWARE DESIGN AND DEVELOPMENT

## 3.1 Component Selection

Microcontroller: ESP32 board with WiFi and low power operation. Sensors: DHT22, MQ135, PIR sensor Supporting components: jumper wires, power supply, optional enclosure.

## 3.2 Circuit Assembly

Steps: 1. Connect DHT22 to GPIO 4. 2. Connect MQ135 to ADC input. 3. Connect PIR output to GPIO 14. 4. Connect ground and power. 5. Validate wiring using multimeter.

## 3.3 Hardware Testing

Initial tests: Power on ESP32 Check serial logs Verify raw sensor readings Adjust PIR sensitivity and delay

# 4. SOFTWARE DEVELOPMENT

### 4.1 Firmware Design

ESP32 firmware tasks: WiFi initialization Sensor polling Data filtering MQTT publishing Deep sleep mode

### 4.2 Data Format and Topic Structure

MQTT topics: homesense/room/temperature homesense/room/airquality Payload example: { sensor: dht22, value: 24.5, unit: C }

### 4.3 Backend Processing Service

Python subscriber service: Validation Conversion to InfluxDB format Logging and storage

# 5. CLOUD AND DATABASE SETUP

### 5.1 MQTT Broker Setup

Mosquitto configuration: Port 1883 Authentication enabled Optional TLS setup

### 5.2 InfluxDB Configuration

Steps: Create database bucket Generate token Set retention policy

# 6. DASHBOARD AND ALERTING

## 6.1 Dashboard Setup

Grafana visualizations: Temperature and humidity trends Air quality thresholds Motion activity timeline

## 6.2 Alert Configuration

Alerts: High temperature for 10 minutes Air quality exceeds threshold Unexpected motion detected

# 7. IMPLEMENTATION STEPS (DETAILED)

## 7.1 Hardware Installation

Steps: Mount sensors away from vents Enclose ESP32 modules Label devices

## 7.2 Firmware Deployment

Steps: Flash firmware Configure WiFi Test MQTT publish Enable auto reconnect

## 7.3 Backend Deployment

Steps: Deploy Python service Run as systemd service Verify end to end flow

# 8. TESTING AND VALIDATION

### 8.1 Functional Testing

Tests: Sensor accuracy comparison MQTT reliability Dashboard updates

### 8.2 Network Reliability Testing

Simulations: WiFi outages Broker restart Offline buffering

### 8.3 Performance Testing

Measurements: Latency Memory usage on ESP32 Database growth rate

## 9. DEPLOYMENT PLAN

### 9.1 Local Deployment

Run stack on Raspberry Pi: MQTT, Python service, InfluxDB, Grafana

### 9.2 Cloud Deployment Option

Cloud setup includes: TLS secured MQTT Remote dashboard access Token based authentication

## 10. MAINTENANCE AND FUTURE WORK

Future enhancements: Battery powered LoRaWAN nodes Mobile configuration app ML based anomaly detection Smart assistant integration

## 11. CONCLUSION

HomeSense provides a scalable smart home monitoring system with detailed implementation across hardware, firmware, backend, and visualization layers.

## 12. REFERENCES

[1] MQTT Specification. [2] ESP32 Technical Manual. [3] InfluxDB and Grafana Documentation.