Procedure and Working of ThermoGuard

1. Problem Identification

Identified the need for predictive maintenance in temperature-sensitive medical and industrial equipment.

Observed that failures due to sensor drift or overheating can cause critical breakdowns and loss of assets.

2. Component Selection and Planning

Sensors: DS18B20 or MLX90614 (accurate, digital temperature sensors).

Microcontroller: ESP32 or Raspberry Pi for processing and transmitting data.

Cloud Backend: Firebase Realtime Database for live storage, alerting, and retrieval.

AI Models: Isolation Forest for anomaly detection, LSTM for time-series trend prediction.

3. System Architecture Design

Sensor captures temperature data from the environment or embedded system.

ESP32/RPi processes the data and sends it to Firebase via Wi-Fi.

Firebase stores real-time data, allowing historical data tracking.

AI Model runs on cloud/edge, analyzing the data to detect abnormal patterns.

Dashboard visualizes the current status of each device.

Alerts (email/SMS) are triggered when predicted anomalies or threshold breaches occur.

4. Hardware Implementation

Sensors are connected to the ESP32 or Raspberry Pi via GPIO pins.

Sensor libraries are used (e.g., Adafruit MLX90614, OneWire for DS18B20).

The microcontroller reads sensor data every few seconds and sends it to Firebase

5. Software and AI Integration

Real-time data is streamed to Firebase and stored as time-series entries.

A Python-based AI model pulls this data:

Isolation Forest flags outliers based on historic patterns.

LSTM predicts future temperature values and detects gradual drift.

When anomalies are detected, Firebase triggers an alert using Firebase Cloud Functions.

6. Dashboard Development

- Built using HTML, JavaScript, and Firebase SDK.
- Displays:
- Real-time temperature readings.
- Device status (Healthy / Warning / Critical).
- Anomaly alerts.
- Maintenance history and logs.

7. Testing and Calibration

Tested using actual and simulated data:

Introduced artificial anomalies like high/low spikes and sensor disconnections.

Adjusted AI thresholds to minimize false positives.

Calibrated sensors using verified digital thermometers.

8. Deployment and Demonstration

- ✓ System deployed in a controlled lab setting.
- ✓ Demonstrated:
- ✓ Real-time monitoring
- ✓ AI-based anomaly detection
- ✓ Instant alerts and visual dashboard response

Working Summary:

- ➢ Sensors collect temperature data from equipment.
- ➢ ESP32/RPi sends data to Firebase continuously.
- ➢ AI Model monitors for abnormal patterns or failure signs.
- ➢ Alerts are generated if a failure is predicted.
- ➢ Dashboard displays live status and allows historical review.
- ➢ Maintenance teams act proactively based on alerts.