**Note:- The web support version has been deployed using render whose service has been suspended currently deploy before using** [**https://web-deployed-medicine-app-logging-verifer.onrender.com/**](https://web-deployed-medicine-app-logging-verifer.onrender.com/)

**📦 Project Overview: Secure Environmental Logging & Traceability in Medicine Packaging**

**🧠 Problem Addressed**

Life-saving medicines are highly sensitive to environmental conditions such as **temperature** and **humidity** during packaging. Any deviation can compromise their effectiveness. However, most packaging systems do not **digitally log these conditions**, nor do they **embed verifiable authenticity** of those conditions on the final product.

**🛠️ Solution Developed**

You built a smart embedded system using **ESP32**, multiple sensors, and a logging+encryption mechanism that:

1. **Measures** the environment (temperature & humidity) at the time of packaging.
2. **Verifies** sterile conditions via a PIR sensor.
3. **Detects** tray presence using an ultrasonic distance sensor.
4. **Generates a SHA256 hash** that encodes packaging metadata.
5. **Logs** all metadata + hash locally in an onboard CSV file.
6. **Prints the hash on the product** (could be printed as a QR or alphanumeric code).
7. Provides a **PC-based Python script** to verify or reverse-lookup the hash for auditing.

**🔧 Hardware Used**

* **ESP32 microcontroller**
* **DHT11 sensor** for temperature and humidity
* **HC-SR04** ultrasonic sensor to detect tray presence
* **PIR motion sensor** to ensure no human movement (sterility)
* **SPIFFS** (ESP32 internal flash) to store logs as logs.csv

**🔐 How the Hash Works**

Each time a valid tray is detected:

* A string is created like:
* LOC123|MACH05|26.0|65.0
* This string is passed to **SHA256** algorithm → a unique 64-character hash is generated
* That hash, alongside the raw data, is stored locally
* The hash can be printed on packaging as a **proof of conditions**

Example:

123456,LOC123,MACH05,26.0,65.0,7d9e2f7e52979a67b6e6e73b86cd49e93a924ccce7cbb8ccac6cf6b4291baf4e

**🖥️ On-PC Script**

A Python script is provided to:

1. **Verify a hash** against provided data (check tampering)
2. **Look up** original metadata if only the hash is known (for auditing)

**💡 Applications**

* Pharmaceutical and vaccine manufacturing
* High-sensitivity biological packaging
* Supply chain traceability systems
* Medical and military logistics