DAFFODIL INTERNATIONAL UNIVERSITY

PROJECT REPORT

Course Code: CSE234
Course Title: Embedded System and IoT Lab
Project title: Smart Vaccine Box Tracker using NodeMCU (ESP8266)

SUBMITTED BY:

MASHUKA BASHAR CHOWDHURY
ID: 221-15-5146

SUMAIYA JANNAT ANIKA
ID: 221-15-5418

NOWSHIN TABASSUM RAHMAN NUHA
ID: 221-15-4795

SECTION : 61_C1
DEPARTMENT OF CSE
DAFFODIL INTERNATIONAL UNIVERSITY

SUBMITTED TO:

MR. AVIZIT NANDI
LECTURER
DEPARTMENT OF CSE
DAFFODIL INTERNATIONAL UNIVERSITY

SUBMISSION DATE : 19 August, 2025

Project Title: Smart Vaccine Box Tracker using NodeMCU (ESP8266)

Course: Embedded Systems and IoT Lab

Team Members:

1. Mashuka Bashar Chowdhury (221-15-5146)
2. Sumaiya Jannat Anika (221-15-5418)
3. Nowshin Tabassum Rahman Nuha (221-15-4795)

Abstract

One of the most delicate medical products is vaccines, which must be stored and transported under rigorous cold-chain conditions. Due to inadequate infrastructure, human monitoring, and a lack of real-time awareness, maintaining this cold chain continues to be a significant challenge in many distant and underdeveloped areas. These flaws frequently result in vaccine deterioration, waste, and decreased efficacy.

We created and deployed a **Smart Vaccine Box Tracker** to solve this problem by fusing cloud computing, QR code technology, and inexpensive IoT hardware. The device, which is based on the NodeMCU ESP8266 microcontroller, uses a DHT11 sensor to continuously check the internal temperature and a NEO-6M GPS module to track its location. After being sent to the Arduino IoT Cloud [1], the data is shown on a real-time dashboard. Each vaccination package has a unique QR code that can be scanned with a smartphone to provide health workers with fast access to current conditions.

This combination of cloud monitoring, IoT sensing, and QR code-based accessibility guarantees a cost-effective, expandable, and intuitive cold-chain management solution. In addition to vaccinations, the system can be modified for medications, perishable foods, and other items that are sensitive to temperature changes. This research shows how intelligent IoT solutions can protect public health and lower avoidable losses in medical supply chains by bridging the gap between technology and healthcare logistics.

1. Introduction

Although vaccines are essential for maintaining global health, the conditions under which they are stored and transported have a significant impact on how effective they are. Manual temperature checks and paper-based records, which are vulnerable to human mistake, delays, and data loss, are frequently used in traditional cold-chain management systems. These restrictions may lead to decreased vaccine effectiveness, resource waste, and challenges in maintaining supply chain accountability.

The **Smart Vaccine Box Tracker** was designed to overcome these obstacles by combining cloud integration, accessible QR codes, and reasonably priced IoT components into a single device. The device's primary functions include tracking the vaccine box's inside temperature and recording its location in real time as it is being transported. Through an easy-to-use online or mobile dashboard, authorized users can remotely monitor conditions thanks to the sensors' data being automatically posted to the Arduino IoT Cloud.

The system's inclusion of QR codes is one of its primary features. Every vaccination box has a unique **QR code** that connects to its dashboard in real time. With just a smartphone scan, health professionals and supply management can now rapidly confirm vaccine conditions, doing away with the need for complicated login processes. The system is not only very effective but also very user-friendly because to the integration of QR technology, especially in field operations where accessibility and speed are crucial.

In conclusion, this concept offers a workable, expandable, and economical way to guarantee vaccine safety. The Smart Vaccine Box Tracker shows how technology may improve public health logistics in both developed and resource-constrained environments by combining cloud services, QR code accessibility, and IoT-based monitoring.

2. Objectives

The Smart Vaccine Box Tracker was developed with the following objectives:

I. **Temperature Monitoring**: To make sure vaccines stay within safe storage ranges, use the DHT11 sensor to continuously monitor and record the vaccine box's internal temperature.

II. **GPS Tracking**: Incorporate the NEO-6M GPS module to give the vaccination box's location updates in real time as it is being transported.

III. **Cloud Connectivity**: For safe storage, display, and remote monitoring, upload sensor data to the Arduino IoT Cloud.

IV. **QR Code Integration**: Make sure each vaccination box has a unique QR code that can be scanned quickly by a smartphone to instantly access the live dashboard.

V. **User Accessibility**: Create a simple, easy-to-use dashboard that supply chain managers and healthcare professionals can understand.

VI. **Cost Efficiency**: To make the system inexpensive and scalable for broad deployment, use widely accessible, low-cost components.

3. System Architecture

3.1 Components Used

I. **Microcontroller**: NodeMCU ESP8266, responsible for processing and Wi-Fi connectivity.

II. **Sensors**:

   A. **DHT11 Sensor**: keeps track of the vaccination box's inside temperature and humidity.

   B. **NEO-6M GPS Module**:provide current geographic location information.

III. **QR Code Module**: Software-based generator that generates distinct QR codes that connect every vaccine box to the cloud dashboard that corresponds to it.

IV. **Cloud Platform**: Arduino IoT Cloud provides real-time display, alerts, and safe data storage.

V. **User Interface**: Dashboards on the web and mobile devices are readily available by scanning the QR code.

VI. **Prototype Hardware**: Breadboard, jumper wires, and regulated power supply for stable operation during testing.

3.2 Block Diagram

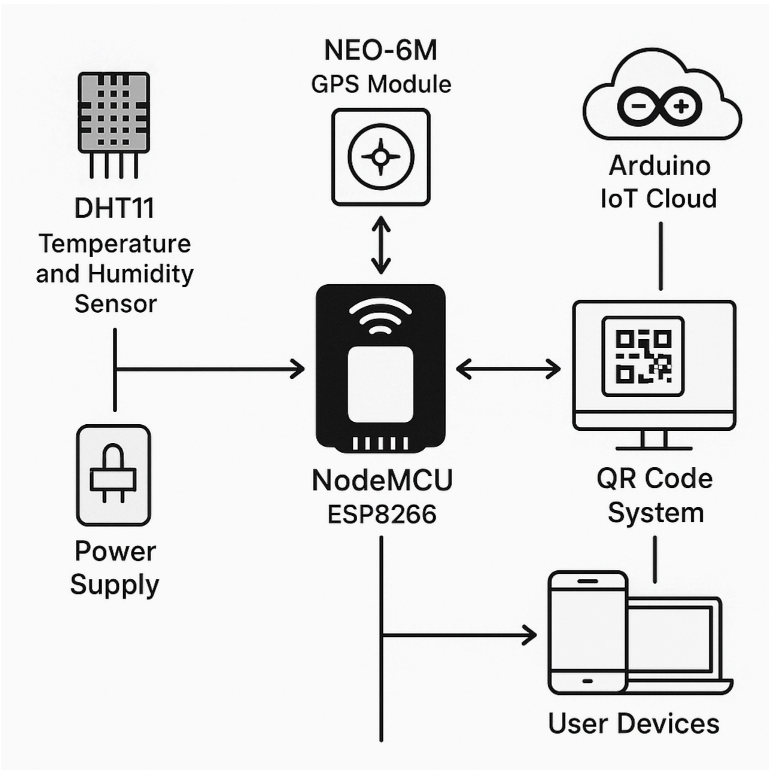

Figure 3.2.1: Overview of the Study's Methodological Workflow.

4. Methodology

4.1 Hardware Design

I. **Microcontroller**: ESP8266 NodeMCU with built-in WiFi (3.3V logic) [2].

II. **Sensors/Modules**: NEO-6M GPS for latitude and longitude information; DHT11 for box temperature and humidity.

Pin Map & Wiring:

I. DHT11 data to D4 (GPIO2), VCC to 3.3 -- 5V (per module), GND to GND.

II. GPS TX to D5 (GPIO14), GPS RX to D6 (GPIO12) via SoftwareSerial; VCC to 5V, GND to GND.

III. All modules share a common ground, and short leads are used to cut down on noise.

Power:

I. The GPS is powered by a 5V adaptor, and the ESP8266 is powered by a controlled 3.3V rail; decoupling capacitors near modules improve stability.

Prototype Platform:

I. All of the components are set up on a breadboard for the purpose of testing and constructing prototypes.

QR Code Integration (Hardware):

I. The vaccination box is physically affixed with a printed **QR code**.
II. Health professionals can access the live dashboard by scanning the QR code with their smartphone's camera, which makes the system easier to utilize in field operations.

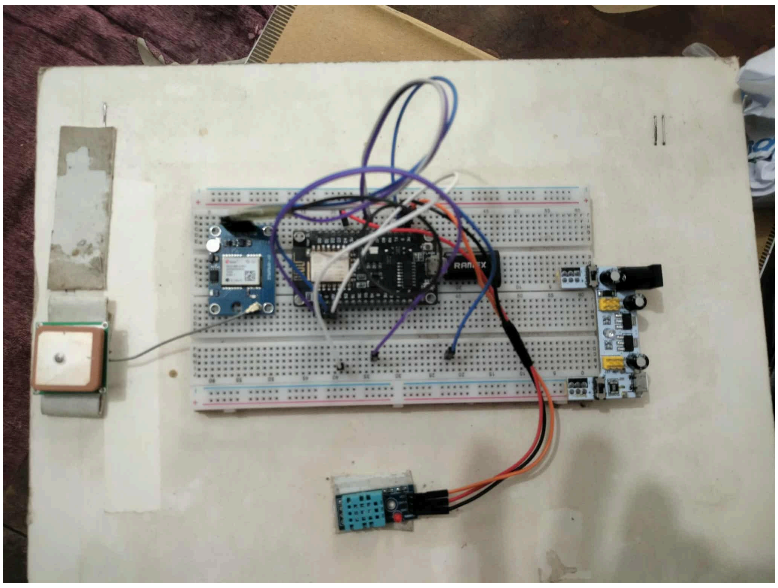

Figure 4.1.1: Prototype built on breadboard along with other components.

4.2 Software Design

I. **IDE & Libraries**: TinyGPSPlus, SoftwareSerial, DHT, ArduinoIoTCloud, Arduino_ConnectionHandler, and ESP8266WiFi are all included with the Arduino IDE.

II. **Sampling and logic**: The main loop parses GPS NMEA until a workable solution is found, publishes every five seconds, and ignores NaN sensor readings.

III. **Cloud Model (Arduino IoT Cloud):**

   A. **Properties**: Temperature (float), Location (lat, lon), Optional alertFlag (bool), lastUpdate (String).

   B. **Dashboard widgets**: Map (location), Gauge (temperature), Time-series graph (history).

QR Code Integration (Software):

I. Each vaccination box's Arduino IoT Cloud dashboard connection is encoded in a unique **QR code**.

II. Without needing explicit login or URL entry, it reroutes the user to the **real-time** dashboard when scanned.

III. For medical professionals on the ground, this guarantees **immediate accessibility.**

4.3 Communication Protocol

I. **Device (Cloud):** Through the secure connection provided by the platform (TLS beneath the hood), the ESP8266 connects to the Arduino IoT Cloud via Wi-Fi.

II. **Update Policy:** A 5-second lag was fixed for reading and transmitting fresh sensor data to the cloud server.

5. Implementation

5.1 Sensor Data Acquisition

To guarantee accurate monitoring of vaccine storage conditions, the system makes use of a number of clever sensors.

Temperature Monitoring:

I. To obtain temperature readings, the system makes use of a DHT11 sensor [3]. The digital output, stability, and ease of integration of this sensor are the reasons it was selected.

II. To offer a more thorough and accurate temperature profile, the vaccination box has many DHT11 sensors placed inside.

III. By combining data from all sensors, an averaging algorithm reduces variations and guarantees reliable outcomes.

IV. Any departure from the predetermined safe temperature limits (such as a range of 2°C to 8°C) immediately results in an alarm to stop spoiling.

Motion Detection:

I. The GPS module is only activated when movement is detected by an inbuilt motion sensor.

II. During idle or stationary storage times, this selective activation greatly extends battery life.

III. To identify instances of unapproved handling or sudden shocks while in transportation, movement data is continuously recorded.

GPS-based Location Tracking:

I. For real-time location updates, the system makes use of an U-Blox NEO-6M GPS module [4].

II. Continuous monitoring and traceability throughout transportation are ensured by periodic coordinate transfer.

III. For end-to-end traceability of the vaccine travel, location data is synchronized with the cloud to generate a comprehensive route history.

Cloud Synchronization:

I. A cellular module periodically uploads all sensor data—temperature, motion, and location—to the cloud.

II. Through the mobile application and other interfaces, this data is made accessible for display and predictive analysis after being safely saved in the cloud.

QR Code Integration in Data Flow:

I. Each vaccine box has **a unique QR code** that links to its real-time cloud dashboard.
II. Without having to manually log into the system, healthcare professionals may quickly receive temperature logs, GPS history, and motion data by scanning the QR code.
III. During field operations, this guarantees prompt evaluation of storage and transit conditions.

5.2 Mobile Application Features

The user's primary control and monitoring interface is the mobile application.

Real-Time Dashboard:

I. Shows real-time motion, temperature, and GPS data in a readable manner.
II. Makes use of clear graphs and charts to visualize trends throughout time.

III. A color-coded alarm system is used to highlight abnormal situations, making it possible to identify problems quickly.

Historical Data Access:

I. Provides comprehensive temperature logs over time, allowing for detailed analysis and compliance checks.

II. Gives thorough temperature logs across time, enabling in-depth analysis and compliance verification.

Manual Controls:

I. Enables users to obtain the most recent data by manually refreshing sensor readings as needed.

II. Gives users the ability to ask for the most recent GPS coordinates whenever they want.

III. Permits the use of simple remote device management commands, including forcing a data sync or restarting the device.

Notification System:

I. Sends push notifications in the event of abrupt temperature changes, which could be a sign of a cold chain breach.

II. Alerts people when something is moved without permission or stored for an extended period of time outside of a safe temperature.

III. Predictive alarms that foresee possible cold chain breakdowns before they happen are based on AI analysis of temperature patterns.

QR Code-Enabled Access:

I. The ability to scan QR codes is integrated into the mobile app.

II. Users can bypass manual URL entering by just scanning the QR code on the vaccine box, which quickly directs them to the related cloud dashboard.

III. Frontline workers may immediately confirm the condition of vaccines thanks to this function, which enhances usability in field operations.

5.3 Security Features

I. Secure connection: Our device uses a secured line (TLS) to transmit data to the Arduino IoT Cloud. Thus, the data is encrypted, making it difficult for unauthorized users to read or alter.

II. Unique device ID: Our board has a unique device ID and secret thanks to Arduino IoT Cloud. This implies that our project can only be published on our device. We cannot be impersonated by others.

III. Verify inaccurate data: Sometimes GPS isn't fixed properly, or DHT11 delivers NaN. We don't upload it in that scenario. In this manner, incorrect values are kept out of our past.

IV. Wi-Fi security: We establish a connection using WPA2. We won't disclose device secrets or Wi-Fi passwords in public codes or screenshots. To prevent critical information from leaking, we additionally limit Serial.print in the final demo.

V. Dashboard privacy: At this time, only our account has access to our Arduino IoT Cloud dashboard, which is a private prototype. We only add persons we can trust if necessary. This aids in preventing access by unauthorized or random people.

6. Results

Live Data Working

Every five seconds, our NodeMCU transmits the temperature value gathered by the sensors and GPS information to the Arduino IoT Cloud. We can observe that the values on the dashboard update automatically without requiring a page refresh or restart.

Dashboard View

I. The vaccination box's current location is displayed on the map widget.

II. The temperature gauge shows the current value, which in the screenshot is approximately 28.6°C.

III. Plotting the line over time, the temperature graph makes it simple to observe variations.

IV. Health professionals can verify readings in real time without logging in manually thanks to the QR code feature, which allows direct dashboard access. Simply scan the printed QR code attached to the vaccination box to activate this dashboard on a smartphone.

Latency

I. Latency problems: Depending on the strength of the Wi-Fi connection, the data is sent to the cloud after being pushed by the device.

II. This delay was acceptable for the class demonstration.

Stability Notes

I. NaN (not a number) values are skipped when DHT11 provides erroneous readings to prevent errors or false alerts from appearing on the dashboard.

II. GPS typically operates more quickly in open air. It could take longer to get a fix indoors, or if the environment is too crowded, it might simply display the latest known location.

Usability

I. Large and readable widgets make it simple to rapidly verify location and temperature in the webview.

II. At a glance, a health worker can understand the state.

III. By eliminating the need to type or navigate menus, the QR code integration enhances usability even further. The dashboard may be accessed instantly by scanning the QR code, which makes it particularly useful for field operations when prompt verification is essential.

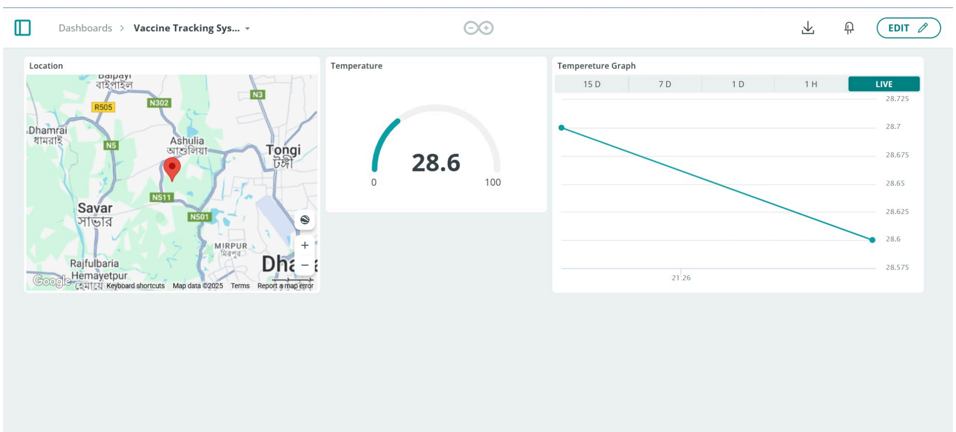

Figure 6.1: Dashboard on web UI

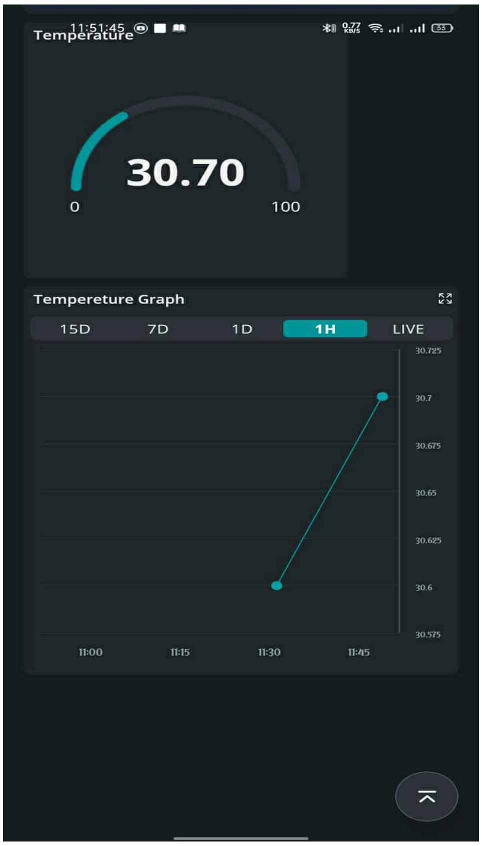

Figure 6.2: Temperature of 1 hour

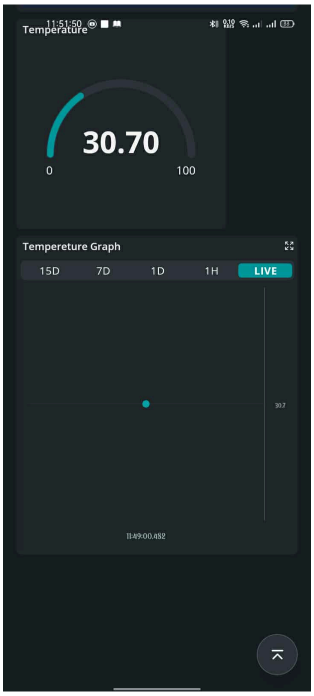

Figure 6.3: Temperature live

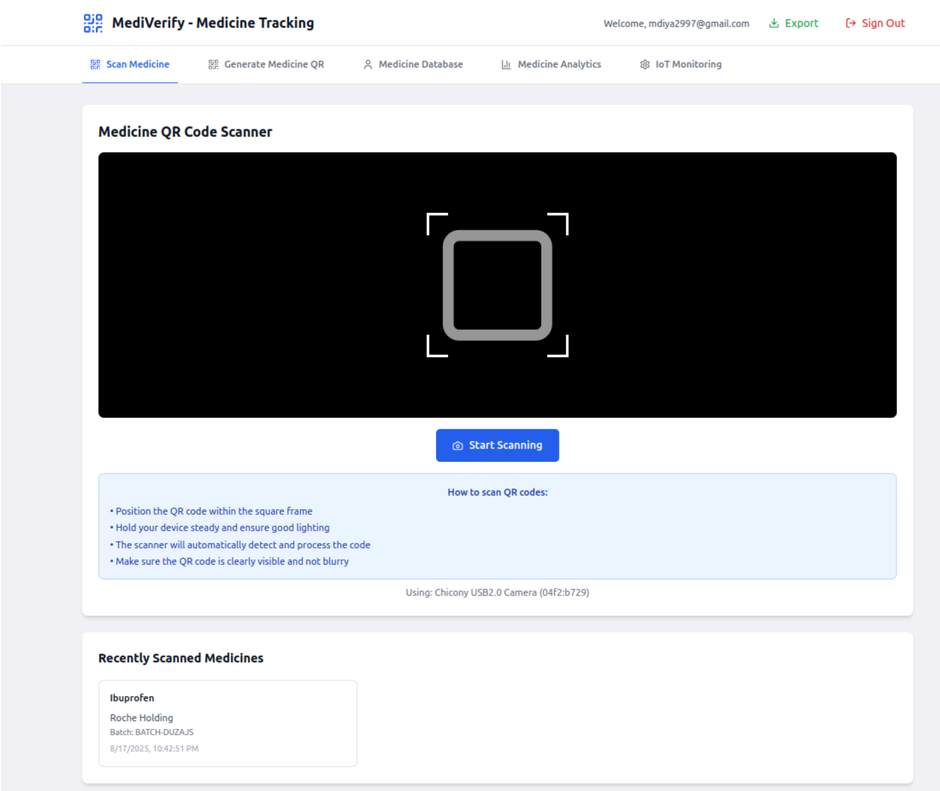

Figure 6.4: Scan Medicine

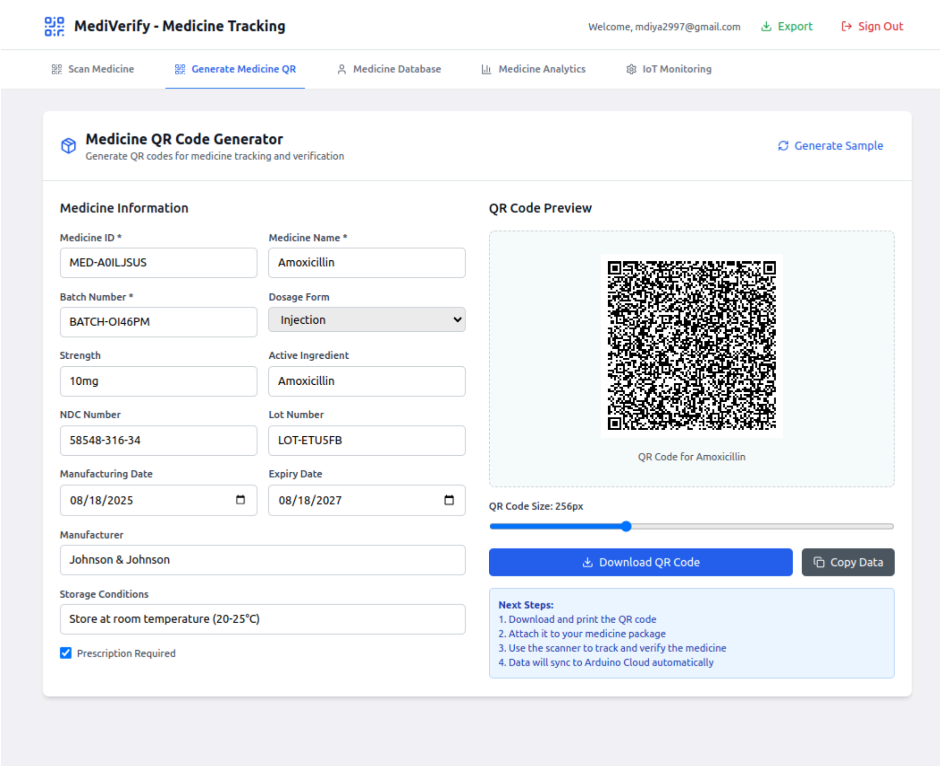

Figure 6.5: Generate Medicine QR

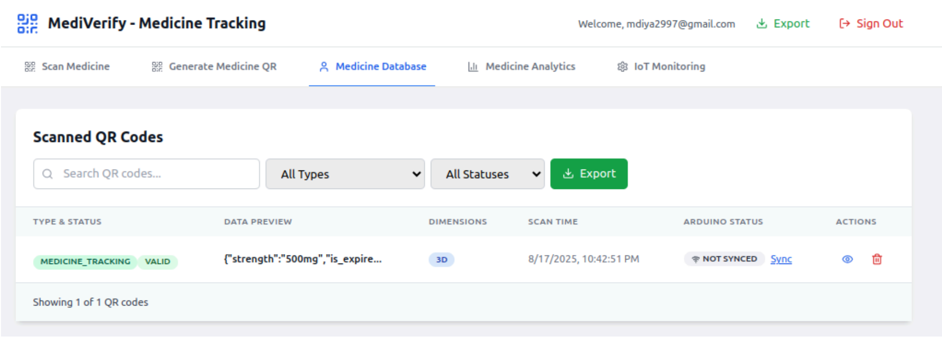

Figure 6.6: Medicine Database

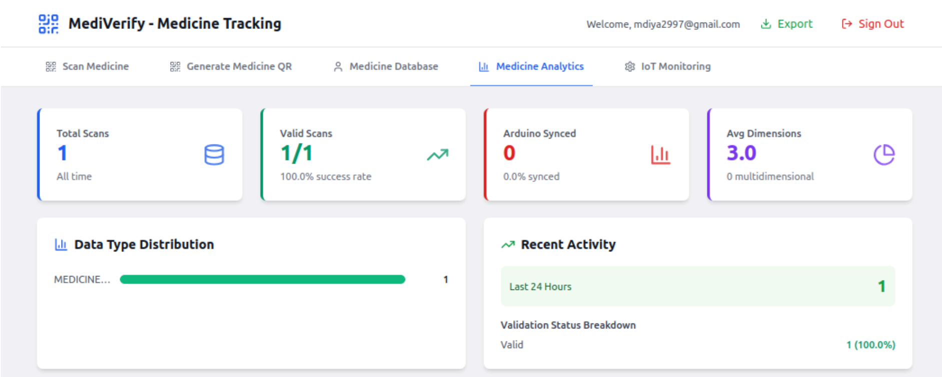

Figure 6.7: Medicine Analytics

7. Challenges and Solutions

Challenge 1: Wi-Fi Drops / Slow Network / Connectivity Issues

I. **Issue:** The prototype occasionally took too long to update the most recent value or failed to do so at all.

II. **Solution:** This problem was resolved by reassembling the ESP8266 and making sure there was a steady Wi-Fi connection.

Challenge 2: GPS Signal Limitation in Indoor Environments

I. **Issue:** Signals inside closed or crowded rooms were not consistently read by the GPS module.

II. **Solution:** The NEO-6M module's signal reception was enhanced by positioning it close to a window or an open area with a view of the sky.

Challenge 3: Too Frequent Updates

I. **Issue:** The dashboard's frequent upgrades occasionally resulted in system problems.

II. **Solution:** Used a preset five-second time limit to push data only when a valid reading was available. The update speed was slowed down and dashboard crashes were prevented.

Challenge 4: DHT11 Sensor Returning Wrong or Undefined Values

I. **Issue:** Undefined (NaN) readings were occasionally generated by the DHT11 sensor.

II. **Solution:** Repositioned the sensor to improve airflow around it. A filter-out method was also included in the code:

```
t = dht.readTemperature();
if (!isnan(t)) { updateCloud(t); }
else { Serial.println("Failed to read DHT11"); }
```

III. This made sure that the cloud didn't receive erroneous readings.

Challenge 5: QR Code Accessibility in Low-Connectivity Areas

I. **Issue:** The live dashboard occasionally took longer to load when scanning the QR code in places with spotty mobile connectivity.

II. **Solution:** A fallback option was added, allowing the QR code to connect to locally stored cached or static data. Future enhancements will guarantee that basic data may be accessed even in the event that real-time cloud updates are not available by implementing an offline buffer.

8. Future Scope

I. **Push Notifications:**

Only the dashboard displays notifications in the present prototype. We plan to incorporate real-time push notifications in the future using a lightweight Telegram bot or a mobile app. The user will immediately receive a pop-up or audio alert if the temperature deviates from the set range or if the box moves unexpectedly.

II. **Improved Temperature Sensor & Calibration:**

Due to its simplicity, the DHT11 sensor might not be accurate enough for use in medical settings. More precise sensors like the DHT22 or DS18B20 will be used in later iterations. To reduce sensor inaccuracy, a straightforward calibration test between ambient temperature and ice-water will also be carried out.

III. **Power Saving & Battery Monitoring:**

Low-power techniques and sleep modes will be used to increase operating time. A voltage divider circuit will be used to track battery percentage, which will then be uploaded to the cloud. In order to conserve electricity, we also intend to test longer update intervals (10–15 seconds) once conditions are steady.

IV. **Offline Buffer & Data Resend:**

The system will temporarily store readings in EEPROM or SPIFFS in the event that Wi-Fi connectivity is lost while being transported. To guarantee that a full history is kept, the data will be uploaded as soon as the internet connection is restored.

V. **Trip Report & Geofencing:**

There will be a designated safe route area. An warning will sound if the box travels outside of the specified area. A PDF or CSV report with the whole temperature graph, GPS path, and time stamps will be automatically generated after delivery. This will facilitate audits and increase accountability.

VI. **Enhanced QR Code System:**

We intend to use dynamic and multi-dimensional QR codes in the future. Richer metadata, including supply chain checkpoints, expiration dates, and shipment IDs, can be stored in these. Additionally, they will enhance traceability and security, making it more difficult to falsify vaccine monitoring information.

9. Conclusion

In this project, we used the NodeMCU ESP8266, DHT11, and NEO-6M GPS to successfully create a Smart Vaccine Box Tracker. Every five seconds, the system updated the Arduino IoT Cloud with temperature and position data. A gauge, map, and time-series graph on the dashboard allowed us to see real-time readings. Invalid GPS coordinates and NaN readings were filtered out to preserve data integrity.

We acquired useful real-world skills through this project, including dashboard design, cloud property configuration, sensor wiring, and GPS data parsing. We were able to put in place a working IoT-based cold chain monitoring system thanks to these abilities. Additionally, we had to overcome issues like indoor GPS delays, Wi-Fi failures, and DHT11 errors using software filtering techniques and hardware modifications.

Our project's incorporation of QR code accessibility is one of its major advances. Health professionals may improve usability and cut down on field delays by instantly accessing the live monitoring dashboard without requiring a manual login by simply scanning the QR code attached to the vaccine box.

The core pipeline functions flawlessly from beginning to end, despite the prototype's shortcomings. Push notifications, better sensors, better power management, offline buffering, and more intelligent QR codes are just a few of the enhancements that might make the system a dependable and expandable cold-chain monitoring solution.

All things considered, this research shows how inexpensive IoT and QR technologies can preserve vaccination efficacy, cut down on waste, and improve healthcare logistics in both wealthy and developing nations.

10. References

[1] Arduino IoT Cloud – Things, Properties, Dashboards & Alerts
https://docs.arduino.cc/arduino-cloud/

[2] Firebase Documentation ESP8266 Arduino Core Documentation (Wi-Fi, networking, examples)
https://arduino-esp8266.readthedocs.io/

[3] Adafruit DHT Sensor Guide & Arduino Library (DHT11/DHT22)
https://learn.adafruit.com/dht

[4] u-blox NEO-6 GPS Modules – Data Sheet / Product Page
https://www.u-blox.com/en/product/neo-6-series