Project Report

Warehouse Cold Storage Monitoring System

A Real-Time Dashboard with Predictive Analytics and Alert Management

Submitted by

Group 4 - E2+PR2 (CSBS 2026)

SRMIST

NAME	Registration Number
Aryan Singh	RA2211042010003
Sharanya Singh	RA2211042010006
Siddharth Dhadwal	RA2211042010011
Ayush Jaiswal	RA2211042010020
Lakshay Shandil	RA2211042010029
Cherry Sharma	RA2211042010051

For: Cognizant SCM Hackathon 2025

Date: September 2, 2025

Table of Contents

Abstract

The integrity of the cold supply chain is paramount for industries ranging from food and beverage to pharmaceuticals. Temperature and humidity deviations can lead to significant financial losses, health risks, and regulatory non-compliance. This report details the design, development, and functionality of the **Warehouse Cold Storage Monitoring System**, a sophisticated, real-time solution engineered to address these challenges.

The system leverages a powerful technology stack including **Python, Streamlit, Pandas, and Altair** to deliver a dynamic and user-friendly monitoring dashboard. It provides a comprehensive overview of four distinct cold storage zones—**Freezer, Chiller, Produce, and Pharmaceutical**—each with customizable environmental thresholds and specialized analytical features.

Through a simulated real-time data generator, the system demonstrates its capability to track key metrics, apply custom alert logic, and provide predictive insights to preemptively identify potential environmental breaches. Key innovations include a multi-tiered alert system with visual, audible, and SMS notifications; zone-specific performance scoring, such as a **Freshness Score for produce** and a **Regulatory Compliance Score for pharmaceuticals**; and detailed historical analysis.

This project showcases a significant leap forward from traditional, reactive monitoring methods, offering a proactive, data-driven, and highly scalable framework for modern warehouse management and the broader supply chain.

2. Introduction

2.1 The Critical Role of Cold Chain Logistics

In an increasingly globalized world, the supply chain that delivers perishable goods—from frozen foods to life-saving vaccines—has become extraordinarily complex. This specialized branch of logistics, known as the "cold chain," relies on the precise, uninterrupted control of temperature and humidity. A failure at any point in this chain can compromise product safety, efficacy, and quality, leading to catastrophic consequences. The global cold storage market is a testament to this, valued in the hundreds of billions of dollars and projected to grow substantially as demand for fresh produce and complex medicines rises.

2.2 Problem Statement

Traditional methods of monitoring cold storage environments are often manual, infrequent, and reactive. They typically involve staff periodically recording temperatures on a clipboard or using basic data loggers that are checked hours or even days later. This approach suffers from several critical flaws:

- **Lack of Real-Time Visibility:** Deviations are often discovered long after they occur, by which time the damage is already done.
- **Human Error:** Manual logging is prone to inaccuracies, missed readings, and data falsification.
- **No Predictive Capability:** There is no mechanism to foresee potential equipment failure or environmental breaches before they happen.
- **Inefficient Data Analysis:** Historical data is often difficult to access, aggregate, and analyze, making it challenging to identify trends or recurring issues.

These deficiencies result in an estimated billions of dollars in product loss annually and pose significant risks to public health.

2.3 Project Objectives

The Warehouse Cold Storage Monitoring System was developed to directly address the shortcomings of traditional methods. The primary objectives of this project are:

1. **To Develop a Centralized, Real-Time Monitoring Dashboard:** Provide a single source of truth for the status of all warehouse cold storage zones.
2. **To Implement Multi-Zone Customization:** Cater to the unique environmental needs of different product categories (frozen, chilled, produce, pharma) with configurable parameters.
3. **To Create a Proactive Alerting System:** Instantly notify personnel of environmental deviations through visual, audible, and mobile (SMS) alerts to enable immediate corrective action.
4. **To Integrate Predictive Analytics:** Utilize historical data to forecast trends and predict potential compliance breaches, shifting from a reactive to a proactive operational model.
5. **To Enhance Data-Driven Decision Making:** Equip managers with tools for historical analysis, performance scoring, and downloadable reports for compliance and operational improvement.

3. Project Overview

3.1 System Synopsis

The Warehouse Cold Storage Monitoring System is a comprehensive software solution designed to simulate the real-time monitoring of a multi-zone cold storage facility. It presents a feature-rich, interactive web-based dashboard where users can oversee the environmental conditions of four distinct zones. The application is built entirely in Python, utilizing the Streamlit framework to create a highly responsive and intuitive user interface without the need for traditional web development expertise. The system's backend is powered by a data generation script that realistically simulates sensor readings for temperature and humidity, including periodic, randomized anomalies to test the system's responsiveness.

3.2 Key Features at a Glance

- **Centralized Dashboard:** A main navigation page provides a high-level "Quick System Status" and easy access to detailed dashboards for each zone.
- **Four Specialized Zones:** The system is pre-configured to monitor a **Freezer, Chiller, Produce, and Pharmaceutical** zone, each with unique default parameters and specialized analytical features.
- **Dynamic and Customizable Thresholds:** Users can dynamically adjust the optimal temperature and humidity ranges, warning buffers, and alert sensitivity for each zone via a sidebar menu.
- **Multi-Modal Alerting:** The system provides instant feedback through on-screen pop-ups, audible browser-based sounds, and critical alerts sent via SMS through a Twilio integration.
- **Advanced Analytics & Scoring:** Beyond simple monitoring, the system calculates performance metrics like a **Freshness Score** for produce and a **Regulatory Compliance Score** for pharmaceuticals, turning raw data into actionable insights.
- **Historical Data Analysis:** Each dashboard includes interactive charts from the Altair library, allowing users to visualize trends over time and analyze historical alert patterns.
- **Data Export:** The system allows for the easy export of data views and alert logs into CSV files for auditing, reporting, and offline analysis.

4. Understanding Warehouse Cold Storage

4.1 What is a Cold Storage Warehouse?

A cold storage warehouse is a facility designed to store perishable goods at controlled, low temperatures. Unlike a standard warehouse, these facilities are essentially large-scale refrigerators or freezers. Their primary purpose is to preserve the quality and extend the shelf life of products such as food, beverages, chemicals, and pharmaceuticals by slowing down deterioration, inhibiting the growth of microorganisms, and maintaining chemical stability.

4.2 Core Components of a Cold Storage System

A typical cold storage facility is a complex system comprising several key components working in concert:

- **Refrigeration System:** This is the heart of the facility. It includes compressors, condensers, evaporators, and refrigerants that work together to remove heat from the storage space and maintain the desired low temperature.
- **Insulation:** To maintain efficiency and prevent temperature fluctuations, the walls, ceiling, and floor of a cold storage facility are built with thick, high-performance insulation materials.
- **Air Handling Units (AHUs):** These units circulate the cooled air throughout the facility to ensure a uniform temperature distribution and prevent "hot spots." They also play a crucial role in controlling humidity levels.
- **Sensors and Controls:** A network of sensors (thermometers, hygrometers) is placed throughout the facility to continuously monitor temperature and humidity. These sensors feed data to a central control system that automates the refrigeration cycle.
- **Insulated Doors and Docks:** Specialized doors and dock seals are used to minimize the loss of cold air and the ingress of warm, moist air during loading and unloading operations.

4.3 The Principle of Zoned Environments

A "one-size-fits-all" approach is ineffective in cold storage. Different products require vastly different environmental conditions. To accommodate this, modern facilities are divided into multiple, independently controlled **zones**. Each zone is a separate room or area within the warehouse that is maintained at a specific temperature and humidity range tailored to the products stored within it.

This project simulates a facility with four common zones:

1. **Freezer Zone:** For products requiring deep-frozen conditions (e.g., frozen meat, ice cream).
2. **Chiller Zone:** For products that need to be kept cool but not frozen (e.g., dairy, fresh meat).
3. **Produce Zone:** For fruits and vegetables, which often require higher humidity in addition to cool temperatures to prevent wilting.
4. **Pharmaceutical Zone:** For medicines and vaccines, which have extremely strict and narrow temperature ranges mandated by regulatory bodies to ensure their efficacy and safety.

By separating products into appropriate zones, a warehouse can ensure the optimal preservation of every item, maximize shelf life, and comply with industry-specific regulations.

5. The Monitoring System in Detail

5.1 Simulated IoT Environment

While the project does not connect to physical hardware, it perfectly mimics a real-world Internet of Things (IoT) environment through its data generation script, `live_data_generator.py`. This script acts as a virtual sensor network, performing the following functions:

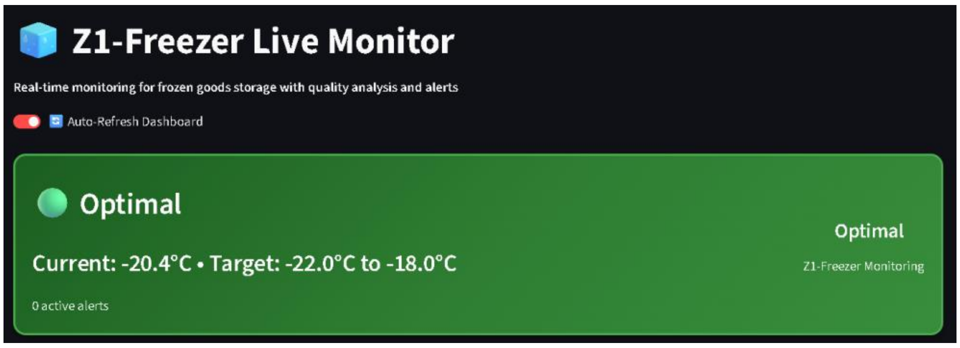
- **Continuous Data Generation:** Every 10 seconds, the script generates a new set of temperature and humidity readings for all four zones.
- **Realistic Value Simulation:** Instead of purely random numbers, the script uses a `generate_realistic_value` function. This function creates plausible data by introducing small, normal drifts from the previous reading, simulating natural fluctuations. It also gently pulls values back towards the optimal range if they drift too far.
- **Controlled Anomaly Injection:** To test the system's alerting capabilities, the script has a `violation_chance` parameter (set to 35%) that intentionally generates readings outside the optimal range. This simulates real-world events like an equipment malfunction or a door being left open.
- **Data Persistence:** The generated data is appended to a CSV file (`simulated_warehouse_data_30min_demo.csv`), creating a persistent historical record that the Streamlit application reads from.

5.2 The Four Pillars: Monitoring Zones

The core of the application lies in its detailed, specialized dashboards for each of the four monitored zones.

5.2.1 Zone 1: The Freezer (Z1-Freezer)

- **Purpose and Stored Goods:** This zone is designed for long-term storage of products that must be kept in a deep-frozen state to prevent spoilage and bacterial growth. Typical goods include frozen meats, poultry, seafood, ice cream, and pre-packaged frozen meals.
- **Operational Parameters:**
  - **Optimal Temperature Range:** -22°C to -18°C
  - **Optimal Humidity Range:** 50% to 70%
  - **Default Warning Buffer:** ±2.0°C
- **Dashboard Features and Analytics (`Z1-Freezer.py`):**
  - **Status Banner:** A prominent, color-coded banner at the top of the page immediately indicates the zone's status (Optimal, Warning, or Critical) using dark, high-contrast backgrounds for readability.



Z1-Freezer Live Monitor

Real-time monitoring for frozen goods storage with quality analysis and alerts

Auto-Refresh Dashboard

Optimal

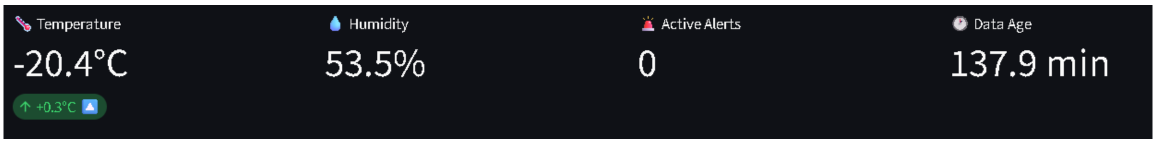
Current: -20.4°C • Target: -22.0°C to -18.0°C

0 active alerts

Optimal

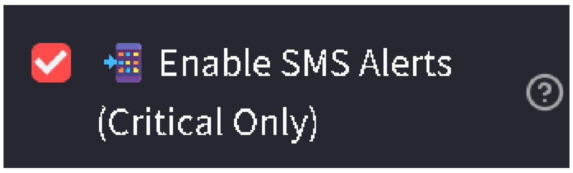
Z1-Freezer Monitoring

  - **Real-Time Metrics:** Key performance indicators (KPIs) are displayed, including the current temperature, humidity, data age, and the number of active alerts.

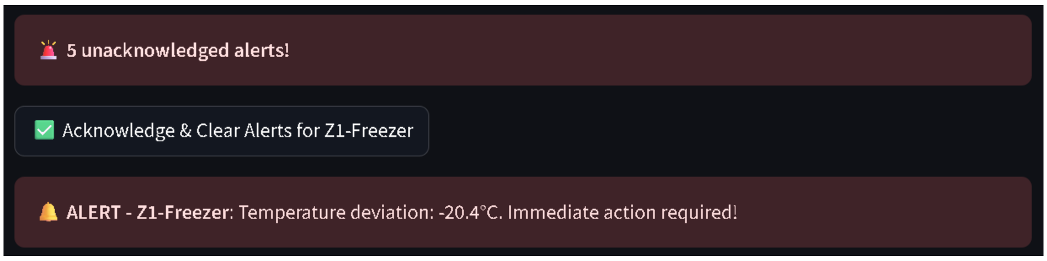


Temperature	Humidity	Active Alerts	Data Age
-20.4°C ↑ +0.3°C	53.5%	0	137.9 min

- **SMS Alert Integration:** The Freezer zone is uniquely configured to send SMS alerts for critical temperature deviations. This feature can be toggled on or off in the sidebar and uses the `sms_sender.py` module.



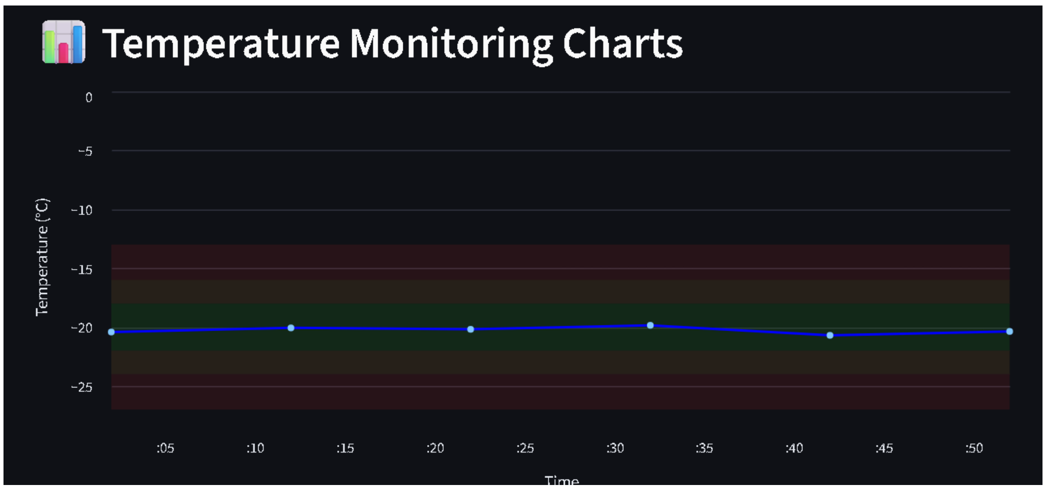
- **Alert Acknowledgement:** To prevent alert fatigue, the dashboard features an "Acknowledge & Clear Alerts" button. This allows operators to confirm they have seen an alert, and the system will only notify them again for *new* deviations.


5 unacknowledged alerts!

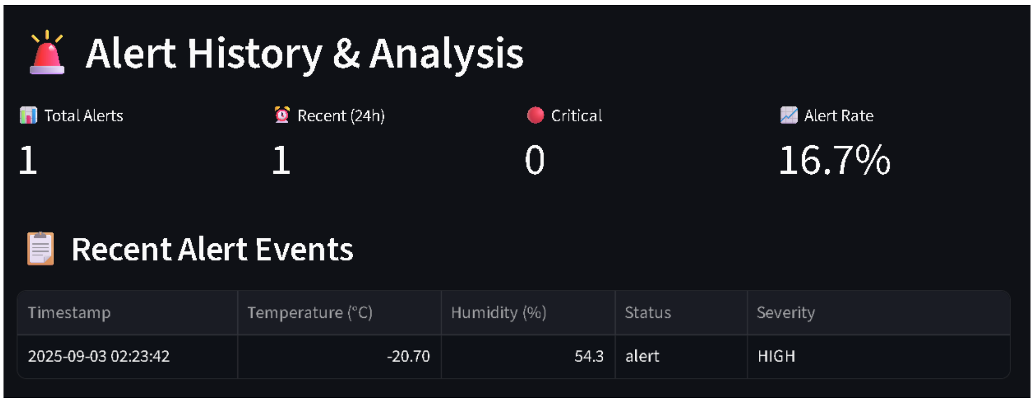
✓ Acknowledge & Clear Alerts for Z1-Freezer

ALERT - Z1-Freezer: Temperature deviation: -20.4°C. Immediate action required!

- **Interactive Temperature Chart:** An Altair chart visualizes the temperature trend over time. Crucially, this chart includes color-coded bands representing the optimal (green), warning (orange), and critical (red) temperature zones, making it easy to see how close the readings are to the thresholds.



- **Alert History and Analysis:** A dedicated section provides a statistical summary of alerts (total, recent, critical) and displays a table of the most recent alert events, including their severity level.



🚨 Alert History & Analysis

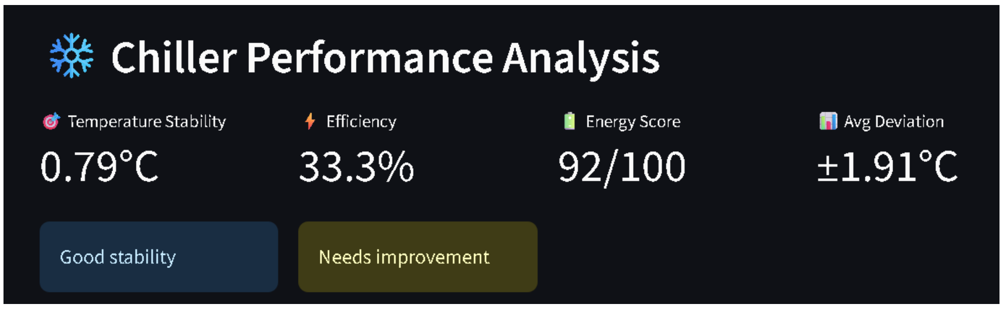
Total Alerts	Recent (24h)	Critical	Alert Rate
1	1	0	16.7%

📋 Recent Alert Events

Timestamp	Temperature (°C)	Humidity (%)	Status	Severity
2025-09-03 02:23:42	-20.70	54.3	alert	HIGH

5.2.2 Zone 2: The Chiller (Z2-Chiller)

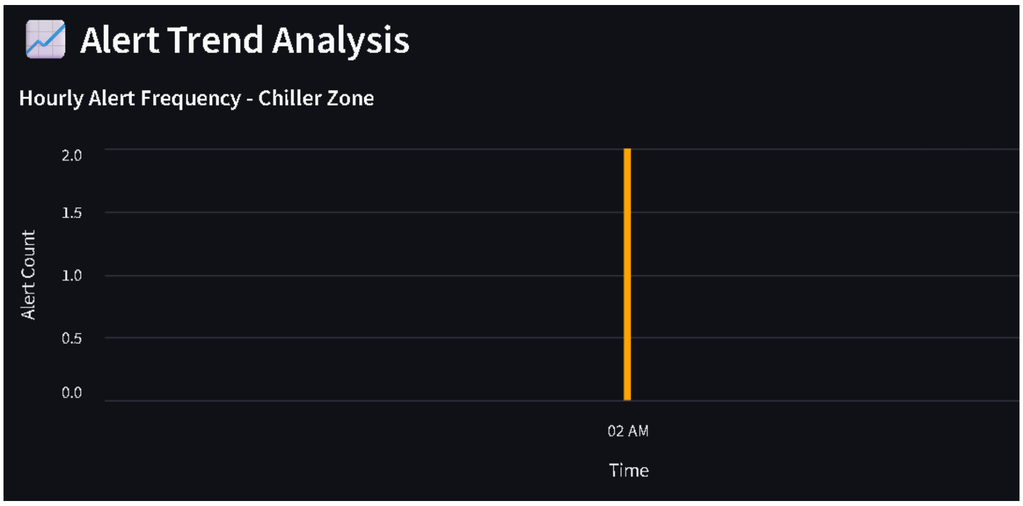
- **Purpose and Stored Goods:** This zone is for short-term storage of fresh, perishable products that require refrigeration but must not be frozen. This includes dairy products (milk, cheese), fresh meat, poultry, deli items, and certain beverages.
- **Operational Parameters:**
    - **Optimal Temperature Range:** 2°C to 5°C
    - **Optimal Humidity Range:** 70% to 80%
    - **Default Warning Buffer:** ±1.5°C
- **Dashboard Features and Analytics (`Z2-Chiller.py`):**
    - **Dynamic Configuration:** Like all zones, the Chiller dashboard has a comprehensive sidebar menu allowing for real-time adjustment of all operational parameters.
    - **Chiller Performance Analysis:** This dashboard includes a unique section to analyze the efficiency of the chiller unit itself. It calculates and displays metrics like:
        - **Temperature Stability (Standard Deviation):** Measures how consistent the temperature is.
        - **Efficiency:** The percentage of time the zone remains within its optimal temperature range.
        - **Energy Score:** An estimated score based on temperature stability.
        - **Average Deviation:** The average difference between the actual temperature and the target midpoint.



Chiller Performance Analysis

Temperature Stability	Efficiency	Energy Score	Avg Deviation
0.79°C	33.3%	92/100	±1.91°C
Good stability	Needs improvement		

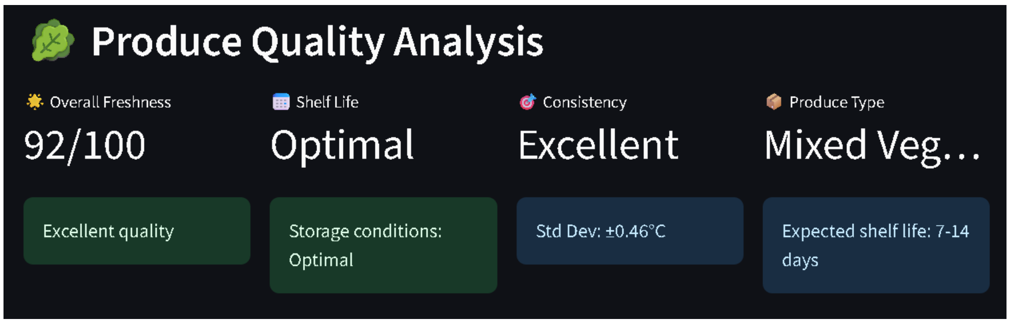
- **Hourly Alert Frequency Chart:** The alert history section includes a bar chart that visualizes the number of alerts per hour, helping to identify patterns (e.g., if alerts frequently occur at a certain time of day).



- **Live Update Control:** The page features an "Auto-Refresh Dashboard" toggle, giving the user control over whether the data updates automatically at a configurable interval or only on a manual refresh.

5.2.3 Zone 3: Produce (Z3-Produce)

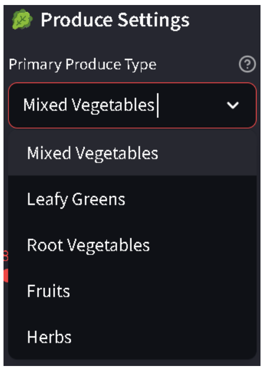
- **Purpose and Stored Goods:** This zone is optimized for fresh fruits and vegetables. These products require not only specific cool temperatures but also high humidity to prevent them from drying out, wilting, and losing freshness.
- **Operational Parameters:**
  - **Optimal Temperature Range:** 5°C to 10°C
  - **Optimal Humidity Range:** 80% to 95% (noticeably higher than other zones)
  - **Default Warning Buffer:** ±2.0°C
- **Dashboard Features and Analytics (`Z3-Produce.py`):**
  - **Freshness Scoring:** The standout feature of this dashboard is the **Produce Quality Analysis**. If enabled, the system calculates an "Overall Freshness" score out of 100. This score is a weighted calculation based on the percentage of time the zone spends within its optimal temperature and humidity ranges, as well as the overall temperature stability.



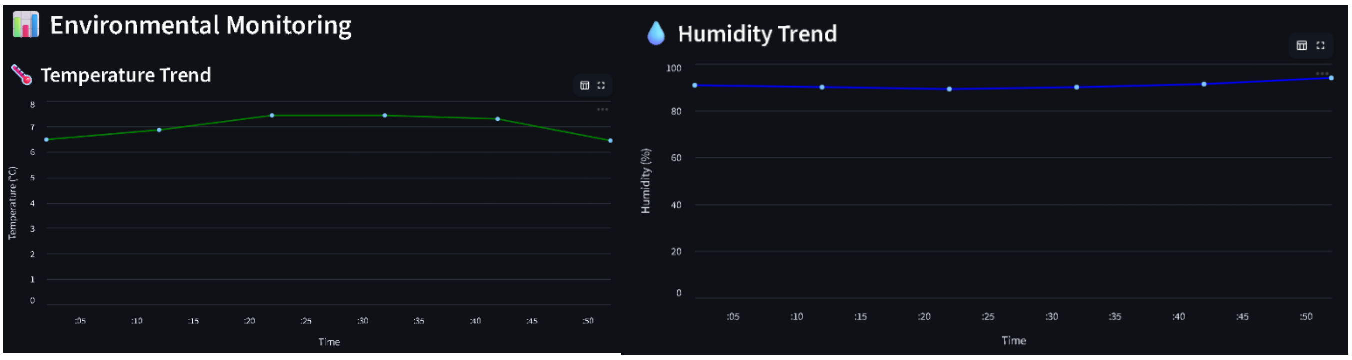
Produce Quality Analysis

Overall Freshness	Shelf Life	Consistency	Produce Type
92/100	Optimal	Excellent	Mixed Veg…
Excellent quality	Storage conditions: Optimal	Std Dev: ±0.46°C	Expected shelf life: 7-14 days

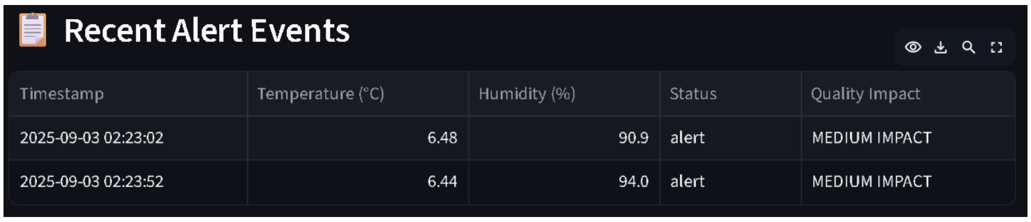
  - **Shelf Life Estimation:** Based on the freshness score, the dashboard provides a qualitative estimate of the produce's shelf life (Optimal, Good, Reduced, or Poor), turning environmental data into a direct business metric.
  - **Produce-Specific Settings:** The sidebar allows the user to specify the primary type of produce being stored (e.g., Leafy Greens, Root Vegetables). The dashboard then provides context-specific information, such as the expected shelf life for that category.



- **Dual Environmental Charts:** Unlike other zones that primarily focus on temperature, the Produce dashboard gives equal prominence to both temperature and humidity trends, displaying them as two separate, large-format Altair charts.



- **Quality Impact Alerts:** The alert history table is customized to show the potential "Quality Impact" (Low, Medium, or High) of each alert, providing more context than a simple severity rating.

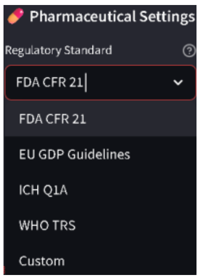


Recent Alert Events

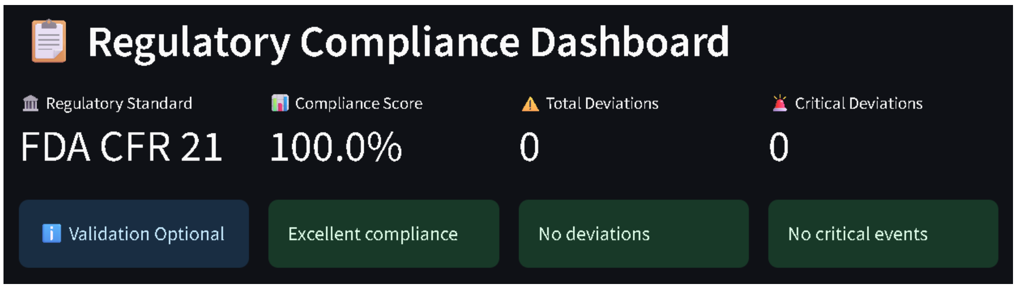
Timestamp	Temperature (°C)	Humidity (%)	Status	Quality Impact
2025-09-03 02:23:02	6.48	90.9	alert	MEDIUM IMPACT
2025-09-03 02:23:52	6.44	94.0	alert	MEDIUM IMPACT

5.2.4 Zone 4: Pharmaceuticals (Z4-Pharma)

- **Purpose and Stored Goods:** This zone is for high-value, highly sensitive pharmaceutical products, including vaccines, biologics, and certain medications. This is the most critical zone, as even minor temperature deviations can render these products ineffective or unsafe.
- **Operational Parameters:**
  - **Optimal Temperature Range:** 2°C to 8°C (a very strict, internationally recognized standard)
  - **Optimal Humidity Range:** 50% to 60%
  - **Default Warning Buffer:** ±1.0°C (a much tighter tolerance)
- **Dashboard Features and Analytics (`Z4-Pharma.py`):**
  - **Regulatory Compliance Focus:** This dashboard is built around the theme of regulatory compliance. The sidebar includes settings to select the relevant regulatory standard (e.g., FDA CFR 21, EU GDP) and to enable options like **Validation Required** and **Automatic Deviation Reporting**.



  - **Compliance Score:** The dashboard prominently features a **Regulatory Compliance Score**. This score is calculated based on the percentage of time the zone remains within its *strict* optimal range, with severe penalties for any critical deviations. This provides an at-a-glance metric for audit readiness.



Regulatory Compliance Dashboard

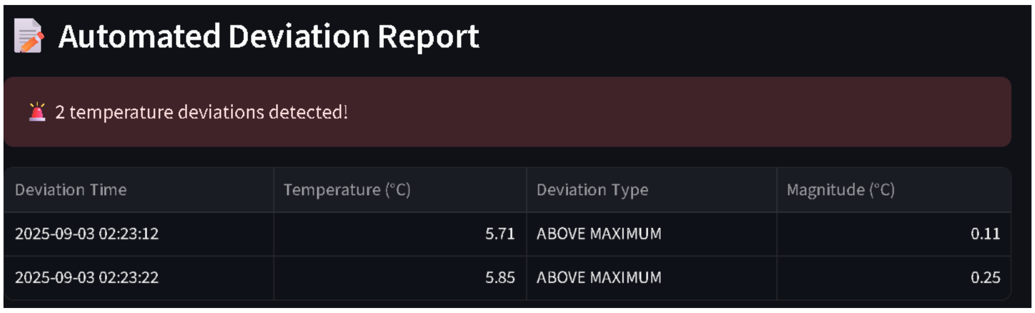
Regulatory Standard: FDA CFR 21 — Validation Optional

Compliance Score: 100.0% — Excellent compliance

Total Deviations: 0 — No deviations

Critical Deviations: 0 — No critical events

- **Automated Deviation Report:** If any temperature readings fall outside the optimal range, the system automatically generates a deviation report table. This table lists the time, magnitude, and type (Above Maximum or Below Minimum) of each deviation, which is crucial for regulatory documentation (e.g., CAPA reports).



Automated Deviation Report

2 temperature deviations detected!

Deviation Time	Temperature (°C)	Deviation Type	Magnitude (°C)
2025-09-03 02:23:12	5.71	ABOVE MAXIMUM	0.11
2025-09-03 02:23:22	5.85	ABOVE MAXIMUM	0.25

- **Critical Alert Logic:** The alert logic is stricter for this zone. Critical alerts are triggered for smaller deviations compared to other zones, reflecting the high sensitivity of the stored products.
- **Compliance Timeline Chart:** An Altair bar chart shows the percentage of compliance for each hour, providing a clear visual history of the zone's performance against regulatory standards.

6. The System's Role in Supply Chain Management

6.1 The Warehouse as a Supply Chain Linchpin

A warehouse is not an isolated entity; it is a critical node in the broader supply chain. It acts as a buffer and a consolidation point, linking producers and manufacturers to distributors and end consumers. In the context of the cold chain, the cold storage warehouse is arguably the most critical link. Any failure within the warehouse has a cascading effect, disrupting the entire chain. Therefore, a system that optimizes warehouse operations directly enhances the resilience, efficiency, and reliability of the supply chain as a whole.

6.2 Ensuring Product Integrity and Quality

The most fundamental contribution of this monitoring system to supply chain management is the **preservation of product quality**.

- **Preventing Spoilage:** By providing real-time alerts for temperature and humidity deviations, the system enables immediate intervention, preventing the spoilage of food products and the degradation of pharmaceuticals. This directly reduces waste and financial loss across the supply chain.
- **Maximizing Shelf Life:** For produce, maintaining optimal conditions is key to maximizing shelf life. The **Freshness Score** feature provides a quantifiable metric that helps in making smarter inventory decisions, ensuring that older stock with a potentially reduced shelf life is shipped out first (First-Expired, First-Out).
- **Guaranteeing Efficacy:** In the pharmaceutical supply chain, the system ensures that temperature-sensitive vaccines and biologics remain effective, safeguarding public health and preventing the immense cost of product recalls.

6.3 Enhancing Inventory Management and Traceability

Effective supply chain management relies on accurate inventory data. This system adds a crucial layer of environmental data to traditional inventory tracking.

- **Quality-Aware Inventory:** By linking environmental data to specific batches or pallets of goods, a warehouse can maintain a "quality-aware" inventory. This means that a batch that was exposed to a minor temperature deviation can be flagged for immediate shipping or closer inspection, even if it's not spoiled.
- **Improved Traceability:** In the event of a product recall, the system's historical data logs are invaluable. They can quickly prove that a specific batch was (or was not) stored under the correct conditions while at the facility, helping to rapidly isolate the source of a problem within the supply chain.

6.4 Strengthening Regulatory Compliance and Reducing Risk

Many industries, especially pharmaceuticals and certain food sectors, are heavily regulated. Failure to comply can result in fines, legal action, and loss of operating licenses.

- **Automated Audit Trails:** The system automatically creates an immutable digital record of environmental conditions. The **Automated Deviation Report** for the pharma zone is a prime example of a feature that drastically simplifies compliance audits.
- **Proactive Compliance Management:** The **Regulatory Compliance Score** acts as a continuous internal audit, allowing managers to address potential compliance issues before they become critical. This proactive stance reduces the risk of costly regulatory penalties.

6.5 Improving Operational Efficiency and Reducing Costs

An efficient warehouse lowers costs for the entire supply chain.

- **Predictive Maintenance:** By analyzing historical data trends (e.g., a chiller's temperature slowly trending upwards over weeks), the system can help predict equipment failures before they happen. This allows for scheduled, preventative maintenance, which is far less costly and disruptive than an emergency breakdown.

- **Energy Optimization:** The **Energy Score** calculated for the chiller zone provides insights into how efficiently the refrigeration equipment is running. High temperature volatility often indicates an inefficient system that is consuming excess energy. Optimizing this reduces the warehouse's operational costs.
- **Labor Efficiency:** Automating the monitoring and logging process frees up staff from tedious manual checks, allowing them to focus on higher-value tasks like inventory management and order fulfillment.

6.6 Increasing Transparency and Trust

A supply chain is a network of partners. Trust and transparency are essential for it to function smoothly.

- **Shareable, Verifiable Data:** The data and reports generated by the system can be shared with upstream suppliers and downstream distributors. This provides verifiable proof that products were handled correctly while under the warehouse's care.
- **Building Confidence:** When a manufacturer entrusts their high-value, temperature-sensitive goods to a third-party logistics (3PL) provider, a state-of-the-art monitoring system like this one builds immense confidence and can be a key competitive differentiator.

7. Advancements Over Traditional Monitoring

7.1 Limitations of Traditional Methods

For decades, cold storage monitoring relied on methods that are now vastly outdated by modern technological standards. These traditional approaches include:

- **Manual Log Sheets:** An employee physically walks through the facility with a clipboard and manually records the temperature from a wall-mounted thermometer at set intervals (e.g., once every 4 hours). This method is labor-intensive, prone to human error, and provides very low-resolution data.
- **Standalone Data Loggers:** Small, battery-powered devices are placed in the storage area to record temperature over time. To access the data, the device must be physically retrieved, connected to a computer, and the data downloaded. This is a step up from manual logs, but it is not real-time. A catastrophic failure could go unnoticed for hours or days until the next scheduled data download.
- **Basic Chart Recorders:** These analog devices use a pen and a rotating paper chart to physically draw a graph of the temperature over time. While providing a continuous record, they offer no remote visibility, no automated alerts, and the data is not easily digitized or analyzed.

7.2 The Modern Approach: A Comparative Analysis

This project represents a paradigm shift from the reactive nature of traditional methods to a proactive, intelligent, and real-time monitoring strategy.

Feature	Traditional Monitoring (e.g., Manual Logs, Data Loggers)	Our Modern Monitoring System	Advantage of Our System
Data Visibility	**Delayed.** Data is only available after manual collection or download.	**Real-Time.** Data is streamed and displayed on the dashboard within seconds of being recorded.	Immediate awareness of current conditions, allowing for instant response.
Alerting	**None or Manual.** An alarm might sound locally, but there are no remote notifications.	**Automated & Multi-Modal.** On-screen, audible, and SMS alerts are triggered instantly based on configurable rules.	Proactive notification to key personnel, regardless of their location, minimizing response time.
Data Analysis	**Difficult and Time-Consuming.** Data must be manually entered or imported for analysis. Trend analysis is a major task.	**Integrated & Interactive.** Historical data is instantly available in interactive charts. The system automatically calculates trends, scores, and statistics.	Effortless identification of patterns, recurring issues, and performance bottlenecks.
Predictive Power	**None.** The system is purely reactive, only showing what has already happened.	**Predictive Analytics.** The system analyzes trends to forecast potential future breaches, enabling preventative maintenance.	Prevents failures before they occur, saving significant costs in lost product and repairs.
Customization	**Rigid.** Alarm thresholds are often difficult to change. No concept of zone-specific logic.	**Highly Flexible.** All parameters for each zone are easily adjustable from the UI without any code changes.	Adapts to different products, seasons, and operational needs on the fly.
Regulatory Compliance	**Labor-Intensive.** Requires manual compilation of logs and reports for audits.	**Automated Reporting.** The system can generate deviation reports and provides a clear compliance score, simplifying audits.	Dramatically reduces the administrative burden of compliance and ensures data integrity.
Accessibility	**On-Site Only.** Data can only be accessed by being physically present at the facility.	**Remote Access.** As a web-based application, the dashboard can be securely accessed from any device with a web browser.	Enables managers to oversee operations from anywhere, at any time.

8. System Architecture and Technology Stack

8.1 Architectural Blueprint

The system is designed with a simple yet powerful and scalable architecture. The flow of data and user interaction can be visualized as follows:

Diagram Breakdown:

1. **Data Generation Layer (`live_data_generator.py`):** This standalone Python script acts as the "Virtual IoT Sensor Network." It runs continuously in the background, generating realistic time-series data for temperature and humidity for all four zones.
2. **Data Persistence Layer (`.csv` file):** The generator writes the data to a shared Comma-Separated Values (CSV) file. This file acts as a lightweight, file-based database, decoupling the data generation process from the data presentation application.
3. **Application Layer (Streamlit Server):** When the user runs the `streamlit run src/Dashboard.py` command, it starts a local web server. This server runs the Python code for the dashboard and all its pages.
4. **Data Processing Layer (Pandas):** The Streamlit application reads the data from the CSV file into a Pandas DataFrame. All data manipulation, filtering, aggregation, and analysis are performed efficiently in memory using the Pandas library.
5. **Presentation Layer (Streamlit UI):** The application uses Streamlit's rich set of commands (`st.metric`, `st.dataframe`, `st.altair_chart`, etc.) to render the data and interactive widgets (sliders, buttons, checkboxes) as an HTML web page.
6. **User Interaction:** The user interacts with the application through their web browser. Every interaction (e.g., moving a slider) triggers a rerun of the Streamlit script, which re-reads the data, re-calculates the analytics with the new parameters, and re-renders the page, providing a seamless, interactive experience.

8.2 Technology Stack Deep Dive

The project's success is built upon a carefully selected stack of open-source Python libraries, as detailed in the `requirements.txt` file.

- **Python (3.8+):** The core programming language used for the entire project. Its simplicity, extensive libraries, and strong data science ecosystem make it the ideal choice.
- **Streamlit (`streamlit`):** The cornerstone of the user interface. Streamlit is a framework that turns Python scripts into interactive web applications with minimal effort. It handles all the complexity of web servers, state management, and widget rendering, allowing the developer to focus purely on the application logic and data. It is used to create the dashboard, sidebar, charts, tables, and all interactive elements.
- **Pandas (`pandas`):** The primary tool for data manipulation and analysis. All the data from the CSV file is loaded into a Pandas DataFrame, which provides a high-performance, easy-to-use data structure. It is used for filtering data by zone or date, calculating statistics (mean, standard deviation), and transforming data for visualization.
- **NumPy (`numpy`):** The fundamental package for scientific computing in Python. It is used by Pandas under the hood for numerical operations. In the `live_data_generator.py` script, NumPy is used specifically for generating realistic data points using functions like `np.random.normal` (for drift) and `np.random.rand` (for triggering anomalies).
- **Altair (`altair`):** A declarative statistical visualization library. Altair is used to create all the interactive and aesthetically pleasing charts in the dashboard. Its declarative syntax makes it easy to build complex visualizations, such as layered charts with different bands for optimal, warning, and critical zones.
- **Twilio (`twilio`):** The communication API used for sending SMS alerts. The `sms_sender.py` module uses the Twilio Python helper library to connect to the Twilio service and send messages for critical alerts, providing an essential out-of-band notification channel.
- **Scikit-learn (`scikit-learn`):** While the current implementation focuses on simulation and real-time display, the inclusion of scikit-learn in the project's stated tech stack and `requirements.txt` file indicates its intended use for the "Predictive Models" feature

mentioned in the README. It would be used to train machine learning models (e.g., regression models to predict future temperatures or classification models to predict the likelihood of an alert).

9. System Workflow: From Sensor to Screen

9.1 A Layman's Guide to the System

Imagine the warehouse has four special rooms (Freezer, Chiller, etc.), and in each room, there's a smart thermometer that checks the temperature and humidity every few seconds.

1. **The Thermometer Reports In:** This "smart thermometer" (our data generator script) writes down its latest reading in a shared digital logbook (the CSV file). It's designed to be realistic, so sometimes it reports a temperature that's a little too high or too low, just like a real freezer might if someone leaves the door open.
2. **The Dashboard is Watching:** A central security guard (our Streamlit dashboard) is constantly watching this logbook.
3. **Checking the Rules:** For each new entry in the logbook, the guard checks it against a set of rules. For example, the rule for the Freezer room is "Stay between -22°C and -18°C." These rules can be changed at any time using a simple control panel on the side of the screen.
4. **Displaying the Status:** The guard updates a big screen on the wall (the dashboard in your browser) showing the current temperature and a color-coded status for each room. Green means "All Good," yellow means "Pay Attention," and red means "Problem!"
5. **Sounding the Alarm:** If a rule is broken, the guard doesn't just turn the status red. It also sounds an alarm on the screen and, if the problem is really serious (like in the Freezer), it sends an instant text message to the warehouse manager's phone.
6. **Learning from the Past:** The guard also keeps a perfect memory of all past readings. The manager can ask the guard to show a chart of the temperature from last week to see if there are any patterns, helping to spot a failing cooling unit before it breaks down completely.

9.2 The Technical Workflow

1. **Data Ingestion:** The process begins with the `live_data_generator.py` script executing an infinite loop. In each iteration, it:
   - o Retrieves the last known temperature and humidity for each zone from the `simulated_warehouse_data_30min_demo.csv` file.
   - o Calls the `generate_realistic_value` function, which uses NumPy to calculate a new reading with random normal drift and a probabilistic chance of a significant deviation (anomaly).
   - o Appends a new row to the CSV file for each of the four zones, including a timestamp, zone ID, temperature, and humidity.
2. **Application Initialization:** A user navigates to the Streamlit application URL. The `Dashboard.py` script executes, creating the main navigation page.
3. **Page Navigation:** The user clicks on a button to monitor a specific zone, for instance, "Monitor Z4-Pharma." Streamlit then navigates to and executes the `Z4-Pharma.py` script.
4. **Data Loading and Processing:**
   - o The `Z4-Pharma.py` script's `load_zone_data` function is called. This function uses Pandas (`pd.read_csv`) to load the entire CSV into memory.
   - o The DataFrame is then filtered to include data only for `zone_id == "Z4-Pharma"`.
   - o The `setup_threshold_configuration` function reads the current values for temperature range, warning buffer, etc., from `st.session_state`. These values are controlled by the interactive sidebar widgets.
5. **Alert Logic Application:** The filtered DataFrame is passed to the `apply_custom_alert_logic` function. This function iterates through each row of the zone's data and adds a new column, `alert_status`, by comparing the row's temperature to the user-defined thresholds from the configuration. The status is set to 'critical', 'alert', 'warning', or 'normal'.
6. **Rendering Visualizations:**
   - o The processed DataFrame is passed to various display functions.
   - o `display_threshold_aware_metrics` uses the last row of the DataFrame to calculate and display the current KPIs using `st.metric`. It also checks the

    `alert_status` of recent readings to trigger the visual `st.error` or `st.warning` notifications.
  - `display_regulatory_compliance` uses the DataFrame to calculate the compliance score and generates the data for the Altair compliance timeline chart.
  - `st.altair_chart` is called with a chart object defined using Altair's syntax, which Streamlit renders as an interactive SVG chart in the browser.

7. **Auto-Refresh Loop:** If auto-refresh is enabled, the script enters a `while True` loop at the end. It pauses for the configured interval (`time.sleep`), clears Streamlit's data cache (`st.cache_data.clear()`), and then programmatically triggers a rerun, which repeats steps 4 through 6 with the newly appended data from the CSV file.

10. Data Simulation Methodology

10.1 The Need for Synthetic Data

In developing a monitoring system, access to a rich, continuous stream of real-world sensor data is ideal but often impractical. Physical sensor installation is costly and time-consuming, and real-world data streams may not contain the specific edge cases (like equipment failures) needed to robustly test the system's alerting and analytical features. Therefore, a high-fidelity synthetic data generator is a crucial development tool. It allows for rapid prototyping, repeatable testing scenarios, and the ability to demonstrate the full range of the application's features without a physical infrastructure.

10.2 Generating Realism: The `live_data_generator.py` Script

The `live_data_generator.py` script was engineered to produce data that is more sophisticated than simple random numbers. Its methodology is based on several principles to mimic the behavior of a real physical system.

1. **Stateful Generation:** The script is not stateless; the next value generated depends on the previous one. It starts by calling `get_last_values()` to read the most recent temperature and humidity for each zone from the CSV file. This ensures a smooth, continuous data stream rather than disjointed random values.
2. **Controlled Drift:** The core of the realism lies in the `generate_realistic_value` function. It introduces a small, random "drift" to the current value using `np.random.normal(0, max_drift * 0.3)`. This simulates the minor, natural fluctuations that occur in any real-world thermal system.
3. **Mean Reversion:** The simulation includes a "gentle pull" mechanism. If a value drifts outside the defined optimal range (but is not a major violation), the script nudges it back towards the optimal range. This models the behavior of a thermostat-controlled system, which is always working to correct small deviations.
4. **Probabilistic Violations:** The most important feature for testing is the simulation of equipment issues. The line `if np.random.rand() < violation_chance:` (with `violation_chance` set to 0.35, or 35%) creates a significant chance that any given reading will be an anomaly. When this occurs, the script generates a value that is

deliberately outside the optimal range by a significant margin (`np.random.uniform(1.5, 4.0)` degrees), simulating events like a cooling unit failure or a door being left ajar.

5. **Environmental Correlation:** The simulation adds another layer of realism by correlating different environmental factors. For example, it introduces a daily sinusoidal temperature variation to mimic the effect of ambient daytime and nighttime temperatures on the system. It also models the inverse relationship between temperature and humidity, where a small increase in temperature often results in a corresponding decrease in relative humidity.

This comprehensive methodology ensures that the data fed into the dashboard is not just random noise, but a believable simulation of a dynamic, real-world cold storage environment.

11. Conclusion

The Warehouse Cold Storage Monitoring System successfully meets all its primary objectives, delivering a robust, real-time, and intelligent solution for modern cold chain management. By leveraging a powerful yet accessible stack of open-source technologies, the project demonstrates that sophisticated monitoring capabilities can be developed rapidly and efficiently.

The system's impact can be quantified through its key features:

- It provides continuous monitoring for **4 distinct and critical warehouse zones**.
- The data generator produces a new data point for every zone every **10 seconds**, resulting in **1,440 readings per zone per day**, a resolution far beyond any manual method.
- The alerting system can distinguish between at least **3 levels of severity** (Warning, Alert, Critical), allowing for a prioritized response.
- The integration of an SMS gateway with a `message_key` in the session state ensures that critical alerts are sent instantly but prevents notification spam, sending only **1 SMS per unique critical event** per session.
- The specialized dashboards translate raw data into actionable business intelligence. The pharmaceutical zone, for example, provides a **compliance score out of 100**, and the produce zone provides a **freshness score out of 100**, directly linking environmental control to financial outcomes and regulatory obligations.

Ultimately, this project serves as a powerful proof-of-concept. It moves beyond the outdated, reactive paradigm of traditional monitoring and presents a modern, proactive framework. By providing real-time visibility, intelligent alerts, and predictive insights, this system empowers warehouse operators to significantly reduce product loss, ensure regulatory compliance, and optimize operational efficiency, directly impacting the safety and quality of the global supply chain.

12. Future Enhancements

While the current system is a comprehensive simulation, its architecture is designed for future expansion and real-world deployment. The following enhancements are recommended as the next steps in its development lifecycle:

- **Integration with Real IoT Sensors:** Replace the data generator script with a module that can ingest data from real-world IoT sensors using protocols like MQTT or HTTP APIs. This would transition the system from a simulation to a production-ready monitoring tool.
- **Advanced ML Models for Prediction:** Fully implement the predictive analytics feature by using the `scikit-learn` library. Historical data could be used to train time-series models (like ARIMA or LSTM) to predict temperature trends several steps into the future and provide a probabilistic risk score for potential breaches.
- **Database Integration:** For production-scale use, replace the CSV file with a robust time-series database like InfluxDB or TimescaleDB. This would improve performance, scalability, and data querying capabilities.
- **User Authentication and Roles:** Implement a user login system with different roles (e.g., Operator, Manager, Administrator) to control access to configuration settings and sensitive data.
- **Automated Mitigation Controls:** In a truly advanced implementation, the system could interface directly with the warehouse's Building Management System (BMS) or HVAC controls to automatically take corrective actions, such as activating a backup cooling unit when a primary unit shows signs of failure.
- **Mobile-Responsive Design and Push Notifications:** Enhance the Streamlit frontend to be fully mobile-responsive and develop a companion Progressive Web App (PWA) that can deliver push notifications for alerts directly to mobile devices.