Title:Visualizing Historical Temperature And Humidity Trends

Phase 2: Submission Document

Abstract:

# This project embraces data-driven decision-making by using IoT and data visualization to showcase historical temperature and humidity trends in public parks. Our goal is to provide valuable insights to park visitors, management, researchers, and nature enthusiasts. We create user-friendly visuals to illustrate long-term environmental patterns, aiding in better decision-making and park services optimization. This project merges technology and nature, offering transparency and accessibility to historical park data, enhancing outdoor experiences, and promoting environmental awareness. It highlights the power of data visualization in creating safer, more enjoyable, and educational parks for all.

Objective:

# The objective is to use data visualization to provide accessible insights into long-term environmental conditions in public parks, benefiting visitors, researchers, and park management. This includes enhancing understanding, enabling informed decisions, optimizing park services, fostering environmental awareness, creating an engaging interface, and aiding pattern recognition. The goal is to make parks safer, more enjoyable, and educational through historical climate data visualization.

Deploying IoT sensors in public park

# 1. \*Identify Objectives\*: Determine the specific goals for deploying IoT sensors. These could include monitoring air quality, tracking visitor traffic, ensuring security, or optimizing irrigation systems.

# 2. \*Select Sensor Types\*: Choose the appropriate sensors for your objectives. For example, you might use environmental sensors for air quality, motion sensors for security, or people counters for visitor tracking.

# 3. \*Network Infrastructure\*: Set up a robust and scalable network infrastructure, such as Wi-Fi, LoRaWAN, or cellular, to connect the sensors to a central system.

# 4. \*Sensor Placement\*: Strategically install sensors in various locations within the park to capture relevant data. For example, air quality sensors should be positioned where pollution might be a concern.

# 5. \*Data Collection\*: Sensors should collect data at regular intervals and transmit it to a central server or cloud platform for analysis. Ensure data security and privacy measures are in place.

# 6. \*Data Analysis\*: Analyze the collected data to derive actionable insights. For instance, analyze visitor traffic patterns to optimize park layout or monitor air quality trends.

# 7. \*User Interfaces\*: Develop user-friendly dashboards and mobile apps for park administrators and visitors to access real-time data and reports.

# 8. \*Maintenance and Power\*: Regularly maintain sensors to ensure they are functioning correctly. Depending on the sensor type, power considerations are vital. Some sensors may be battery-operated, while others might require a continuous power source.

# 9. \*Security and Privacy\*: Implement robust security measures to protect the data and the IoT network. Be transparent about data collection and usage to address privacy concerns.

# 10. \*Scaling\*: As the park's needs evolve, be prepared to scale the IoT network and add more sensors or features.

# 11. \*Feedback and Improvement\*: Continuously gather feedback from park users and administrators to make improvements and refine the IoT system.

# 12. \*Regulatory Compliance\*: Ensure compliance with local regulations and data protection laws, especially when dealing with data collected from public spaces.

Protocols:

# IoT environmental sensors can be deployed in the most inhuman locations, allowing data to be collected without the need for regular visits by humans. These sensors can be accessed over the internet or over other wireless IoT protocols, ideally miles away from the target location.

# One of the challenges in deploying IoT sensors for environmental monitoring is the lack of internet connectivity in most places

# One of the challenges in deploying IoT sensors for environmental monitoring is the lack of internet connectivity in most places, or the exorbitant cost of providing internet infrastructure in those places. For example, in wildfire monitoring in the rainforest, the chances of having internet connectivity are nearly zero unless you are relying on satellite connectivity, which also adds the question of power and costs.

# IoT-based environment monitoring needs to be designed with sustainability at its core. This means it should be easily accessible, support long-distance communication, consume low power, have long usable life span, be durable, and be cost-effective. Firmware updates should also be easy to apply over the air if possible as firmware updates are essential for large scale deployment of connected devices. One IoT protocol that fits these requirements is low-power, wide-area network (LPWAN).LPWAN is often regarded as the most well-suited to environmental monitoring. Sensors that use LPWAN can stay in a location for long periods and send data over long distances. LPWAN is also energy efficient and scalable; a battery on an LPWAN based remote device could potentially last up to 10 years without the need for a recharge. The low throughput is ideal for many industrial applications. Three prominent LPWAN protocols are

# LoRa (LoRaWAN)

# SigFox

# Weightless

# Each of these technologies offers its benefits, although comparing each of these is beyond the scope of this post. EDN released an excellent whitepaper that compares each of these protocols. Note page 4 in this PDF in particular, which includes a table with frequency bands, channel width, range, transmit power, packet sizes (minimal or maximal), downlink and uplink data rates, maximum number of connected devices, topology, roaming capability, and status for these protocols.

Program:

import time

import datetime

import random # Replace with actual sensor library

# Function to read sensor data (replace with your sensor library)

def read\_sensor\_data():

temperature = random.uniform(15.0, 30.0) # Replace with actual temperature reading

humidity = random.uniform(30.0, 70.0) # Replace with actual humidity reading

return temperature, humidity

# Function to log data to a file

def log\_data(temperature, humidity):

timestamp = datetime.datetime.now()

log\_entry = f"{timestamp}: Temperature={temperature}°C, Humidity={humidity}%\n"

with open("environmental\_data.txt", "a") as file:

file.write(log\_entry)

# Main loop

while True:

try:

temperature, humidity = read\_sensor\_data()

log\_data(temperature, humidity)

print(f"Data logged: Temperature={temperature}°C, Humidity={humidity}%")

time.sleep(3600) # Log data every hour

except Exception as e:

print(f"Error: {str(e)}")

time.sleep(60) # Wait for a minute before retrying in case of an error

Platform:

# 1. \*Arduino and Raspberry Pi\*: These popular development boards are commonly used for DIY environmental monitoring projects. They can interface with various sensors and send data to a local server or the cloud.

# 2. \*IoT Platforms\*: Internet of Things (IoT) platforms like AWS IoT, Azure IoT, or Google Cloud IoT provide tools and services to collect, store, and analyze data from environmental sensors. They often support a wide range of devices and can scale for larger projects.

# 3. \*Rugged Data Loggers\*: For harsh or remote environments, specialized data loggers like those from Campbell Scientific or Onset are used. They are built to withstand extreme conditions and can record data for extended periods.

# 4. \*Wireless Sensor Networks\*: These networks use wireless sensor nodes to collect and transmit data. Examples include Zigbee, LoRaWAN, or NB-IoT, which are suitable for monitoring applications across large areas.

# 5. \*Environmental Monitoring Systems\*: Companies like Davis Instruments and Airmar provide pre-built environmental monitoring systems for weather, ocean, or air quality monitoring.

# 6. \*Remote Sensing Satellites\*: For large-scale environmental monitoring, satellites can capture data on a global level. Organizations like NASA and the European Space Agency have environmental monitoring satellites.

# 7. \*Smartphone Apps\*: Some citizen science projects use smartphone apps to collect environmental data. For example, apps can crowdsource data on air quality or wildlife observations.

# 8. \*Weather Stations\*: Personal weather stations, like those from companies such as AcuRite and Ambient Weather, offer a range of sensors for weather and environmental monitoring.

# 9. \*Industrial SCADA Systems\*: In industrial settings, Supervisory Control and Data Acquisition (SCADA) systems are used for environmental monitoring, especially for processes that generate emissions or require precise control.

# 10. \*Web-Based Portals\*: Some organizations provide web-based portals for citizens and scientists to contribute and access environmental data, such as air quality indices and weather information.

# The choice of platform depends on factors like the scale of monitoring, budget, specific environmental parameters to measure, and the need for remote or real-time data access.

Conclusion:

# Incorporating data visualization techniques for historical temperature and humidity trends offers several benefits: improved data interpretation, enhanced communication, long-term planning, early warning systems, and educational value. These visualizations act as a bridge between data and actionable insights, contributing to a more sustainable and resilient future.

Github link:

# https://github.com/NITHYASHANMUGAM66/nithya-shanmugam.git