ENVIRONMENTAL MONITORING

Design Thinking Approach:

1. Empathize:

- \*Identify Stakeholders:\* Recognize key stakeholders, including government bodies, environmental organizations, scientists, and the general public, to grasp their environmental monitoring needs and concerns.

User Research: Undertake surveys, interviews, and workshops to amass insights into specific environmental issues and data requirements across diverse user groups.

2. Define:

- Problem Formulation: Articulate a well-defined problem statement by synthesizing insights from the empathize phase. For instance, "How can we develop a scalable and user-friendly environmental monitoring system to effectively address climate change and pollution?"

-Constraints Identification:\* Consider budgetary limitations, technological boundaries, and regulatory prerequisites that may impact the design.

3. Ideate:

- Brainstorm Solutions: Foster inventive thinking to generate an extensive array of ideas for monitoring systems and tools.

- \*Idea Prioritization:\* Evaluate and rank ideas based on criteria like feasibility, potential impact, and alignment with user needs.

4. Prototype:

- \*Develop a Prototype:\* Construct a simplified prototype of the environmental monitoring system for testing and iterative enhancement.

- \*User Testing:\* Solicit feedback from stakeholders and users to refine the prototype and effect necessary improvements.

5. Test:

- Pilot Implementation: Execute a small-scale pilot project to gauge the monitoring system's effectiveness in a real-world context.

- Continuous Feedback Collection: Continuously gather feedback from users and stakeholders during the pilot phase to pinpoint issues and needed refinements.

6. Implement:

- Scaling Up: If the pilot proves successful, blueprint the full-scale implementation of the environmental monitoring system.

- Collaboration: Form partnerships with pertinent organizations and agencies to ensure data sharing and cooperative efforts.

7. Evaluate:

- Monitoring Impact: Continuously assess the monitoring system's influence on environmental awareness, policy decisions, and positive behavioral changes.

- Iterative Enhancement: Utilize the feedback and data collected to enact ongoing improvements to the system.

8. Communicate:

- Disseminate Findings: Share the discoveries and outcomes of the environmental monitoring system with the public, policymakers, and other stakeholders to boost awareness and encourage environmentally responsible actions.

Sensors for Virtual Environments:

Temperature Sensor: This sensor is designed to gauge fluctuations in temperature within the environment, a fundamental aspect of climate change monitoring.

2. Humidity Sensor: Humidity sensors specialize in measuring moisture levels in the air, rendering them invaluable for tracking humidity's influence on agriculture and meteorological phenomena.

3. Air Quality Sensor: These sensors are engineered to detect key air pollution parameters, encompassing particulate matter (PM2.5 and PM10), carbon monoxide (CO), and volatile organic compounds (VOCs). They assume a pivotal role in monitoring air quality.

4. Light Sensor:Light sensors are adept at quantifying ambient light levels, proffering insights into daylight patterns, plant growth cycles, and the repercussions of light pollution.

5. Sound Sensor: These sensors adeptly capture ambient noise levels, facilitating the surveillance of noise pollution and its ramifications for both wildlife and human communities.

6. Gas Sensors:Tailored to your specific needs, gas sensors possess the capability to identify and quantify distinct gases such as methane, ozone, or nitrogen dioxide.

7. GPS Module:Should geographic precision be imperative, a GPS module can furnish precise geographical coordinates.

8. Water Quality Sensors: For the meticulous examination of bodies of water, sensors are equipped to evaluate critical parameters such as pH levels, dissolved oxygen concentrations, turbidity, and electrical conductivity.

9. Soil Moisture Sensor: These sensors prove instrumental in agricultural applications and soil health assessments by measuring soil moisture content.

1. Motion Sensors: In specialized instances, motion sensors like Passive Infrared (PIR) sensors can discern the presence of animals or humans, serving purposes ranging from wildlife monitoring to security applications.

ENVIRONMENTAL MONITORING USING RASPBERRY PI:

Environmental monitoring using a Raspberry Pi involves setting up sensors to track various parameters such as temperature, humidity, air quality, and more. By connecting these sensors to the Raspberry Pi, you can collect, process, and visualize the data. This setup allows you to monitor and analyze the environmental conditions in real time, enabling proactive measures for maintaining optimal conditions and ensuring sustainability.

1. Hardware setup: Connect sensors (e.g., temperature, humidity, and air quality) to your Raspberry Pi.

2. Software installation: Install necessary libraries for sensor data collection and processing.

3. Data collection: Configure the Raspberry Pi to collect real-time data from the connected sensors.

4. Data processing: Write scripts to process the collected data for analysis and visualization.

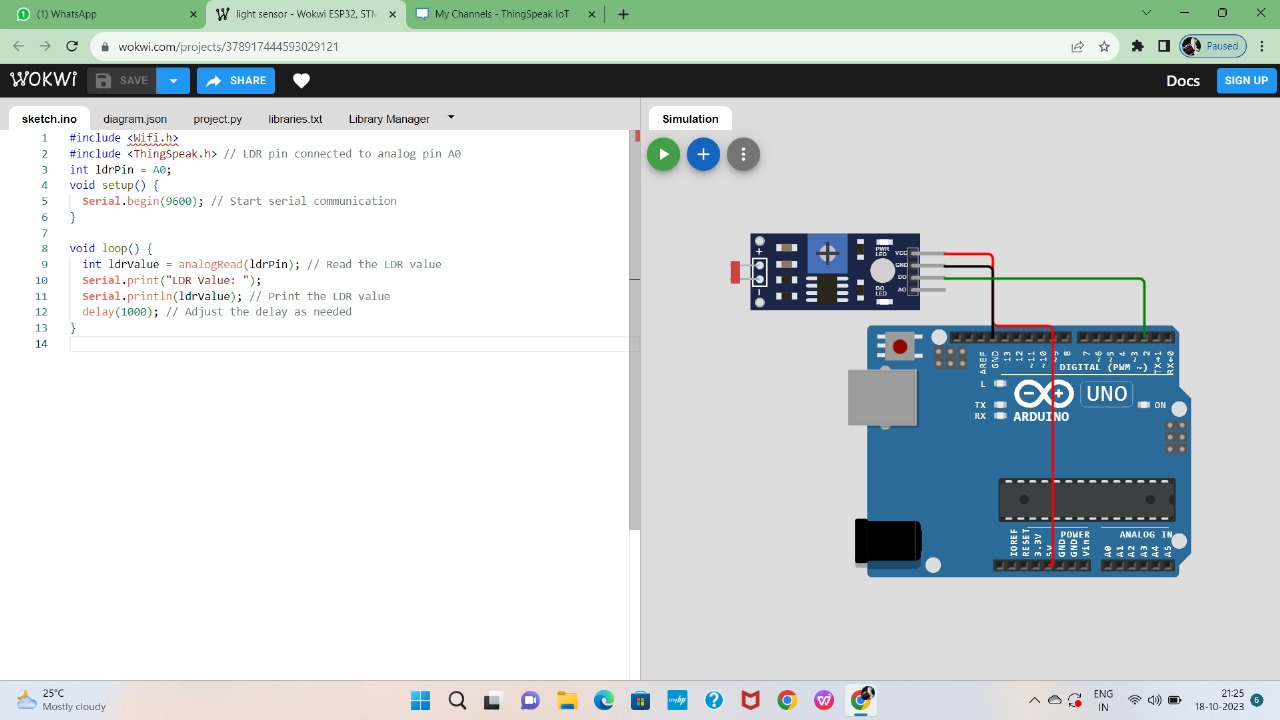
5. Data visualization: Utilize tools like graphs or charts to represent the data for easy understanding.

6. Alert system: Set up alerts or notifications for critical environmental changes or thresholds.

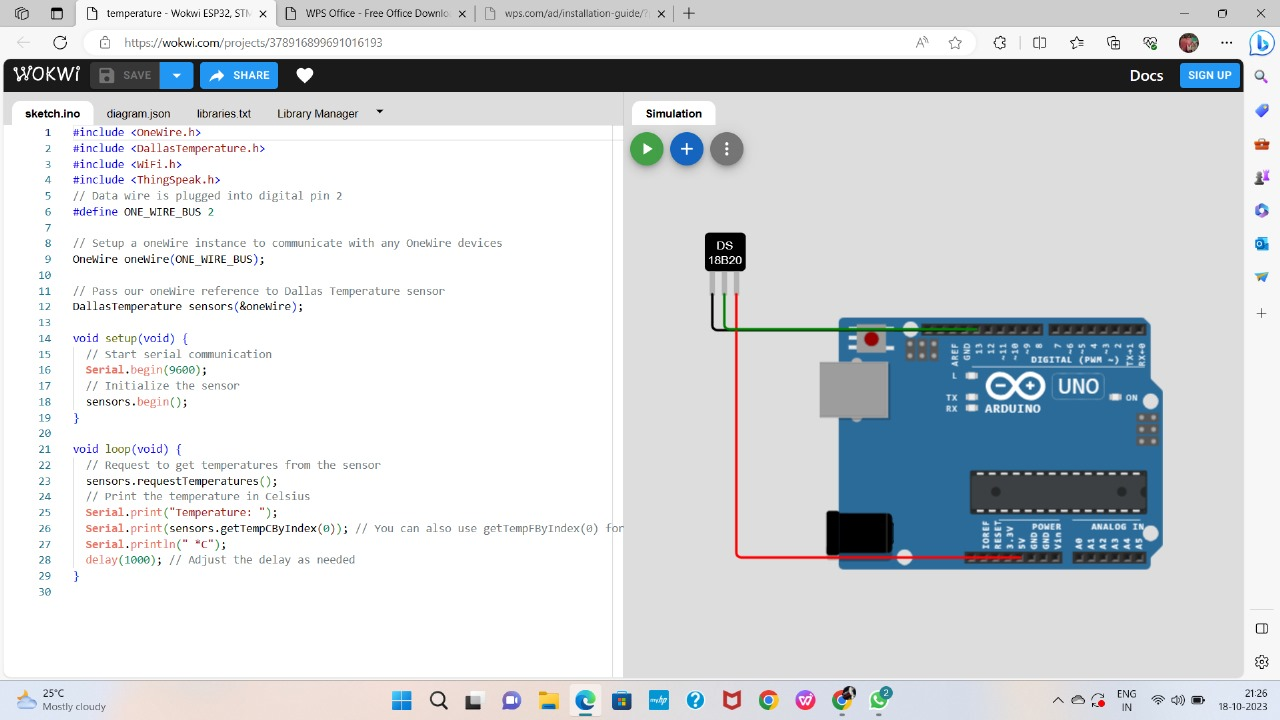
7. Remote access: Enable remote access to monitor the data from anywhere and ensure the system’accessibility.

SCREENSHOTS:

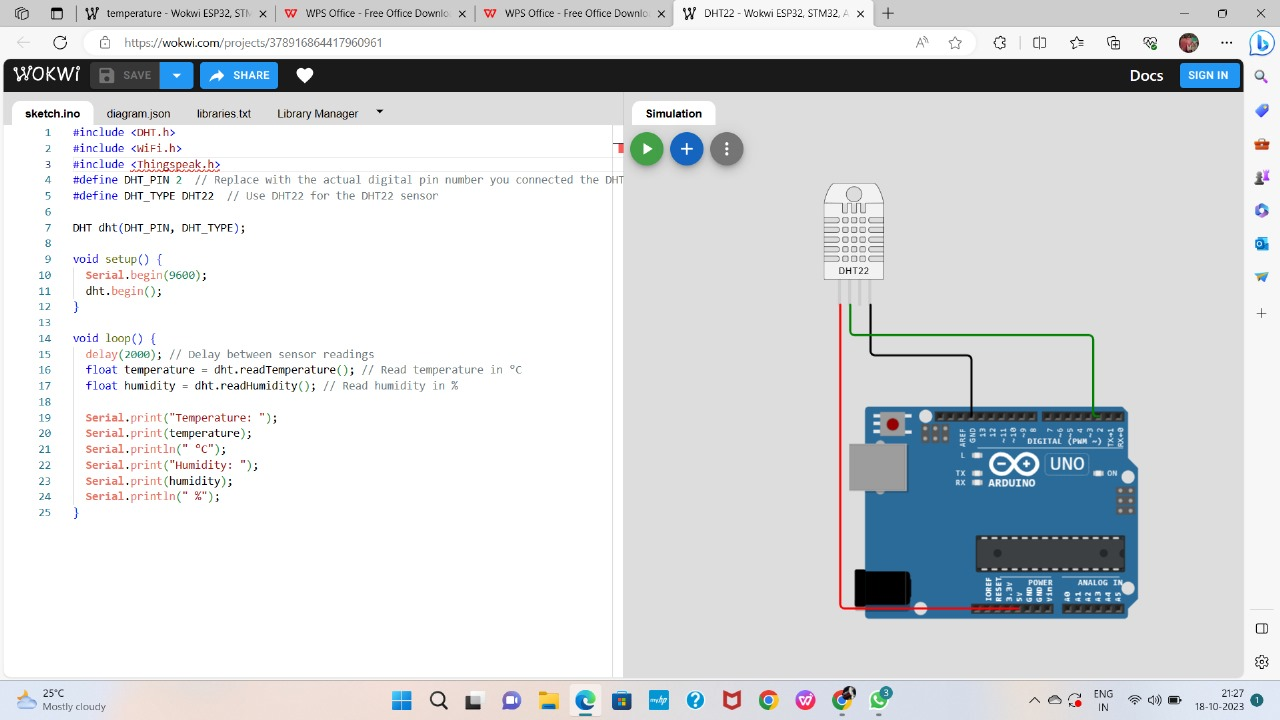
LIGHT SENSOR



TEMPERATURE SENSOR:



HUMIDITY SENSOR:



PYTHON SCRIPT CODE FOR SENSORS:

Python script for sharing data from temperature and humidity sensor

import requests

import time

import json

thingspeak\_url = "https://api.thingspeak.com/update"

api\_key = "UZUC7BGXH0VMGKJX"

ssid = "Wokwi-GUEST"

password = ""

DHT\_PIN = 15

TRIG\_PIN = 13

ECHO\_PIN = 12

def get\_distance():

  from machine import Pin

  import dht

  dht\_sensor = dht.DHT22(Pin(DHT\_PIN))

  while True:

    try:

        dht\_sensor.measure()

        temperature = dht\_sensor.temperature()

        humidity = dht\_sensor.humidity()

        distance = get\_distance()

        print("Temperature: {:.2f}°C, Humidity: {:.2f}%, Distance: {:.2f} cm".format(temperature, humidity, distance))

        data = {

            "api\_key": api\_key,

            "field1": temperature,

            "field2": humidity,

            "field3": distance

        }

        response = requests.post(thingspeak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Data sharing for light sensor:

import requests

import time

import json

thingspeak\_url = "https://api.thingspeak.com/update"

api\_key = "UZUC7BGXH0VMGKJX"

Channel\_id=2310798

ssid = "Wokwi-GUEST"

password = ""

DHT\_PIN = 15

TRIG\_PIN = 13

ECHO\_PIN = 12

def get\_distance():

  from machine import Pin

  import dht

  dht\_sensor = dht.DHT22(Pin(DHT\_PIN))

  while True:

    try:

        dht\_sensor.measure()

        temperature = dht\_sensor.temperature()

        humidity = dht\_sensor.humidity()

        distance = get\_distance()

        print("Temperature: {:.2f}°C, Humidity: {:.2f}%, Distance: {:.2f} cm".format(temperature, humidity, distance))

        data = {

            "api\_key": api\_key,

            "field1": temperature,

            "field2": humidity,

            "field3": distance

        }

        response = requests.post(thingspeak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)