PHASE 5

SMART WATER MANAGEMENT

TEAM MEMBERS:

G Bharathi 412721106007

S Kalpana 412721106022

T Raghul 412721106036

M Vignesh 412721106051

AGENDA

WORK ASSIGNED:

- Consider incorporating machine learning algorithms to analyze water consumption patterns and provide conservation suggestions.
- Start building the IoT flood monitoring and early warning system. Deploy IoT sensors (e.g., water level sensors) in flood-prone areas and configure them to measure water levels. Develop a Python script on the IoT sensors to send collected water level data to the early warning platform.
- Building the project by developing the data-sharing platform. Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water consumption data. Design the platform to receive and display water consumption data from IoT sensors and promote water conservation efforts.

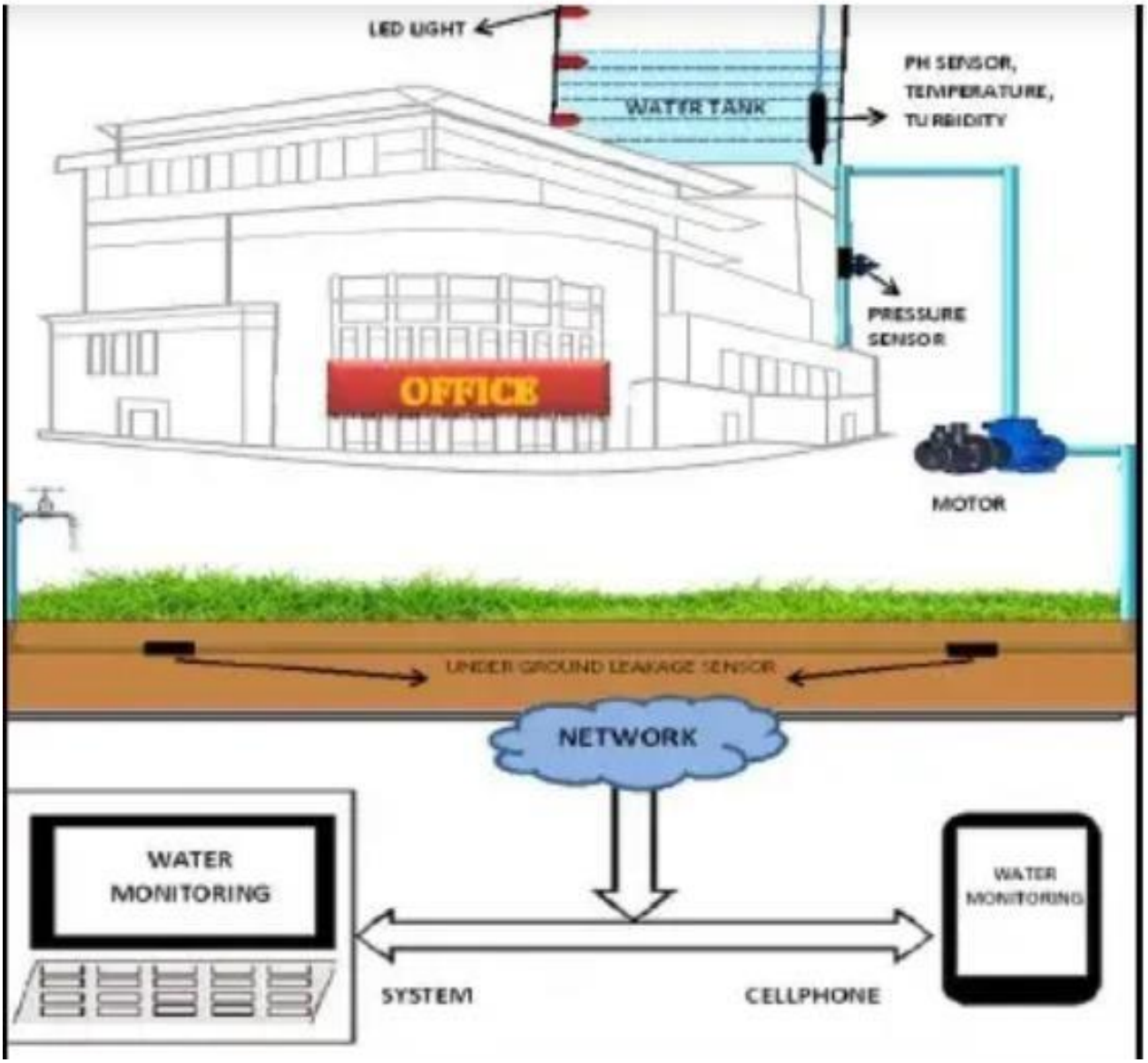
OBJECTIVE:

Design and implement an integrated IoT-based solution for smart water management that delivers real-time water level monitoring in tanks via a user-friendly mobile application. The project aims to provide remote users with accurate and instant access to water levels while offering control over a motor for turning the system ON or OFF. Additionally, the incorporation of a machine learning algorithm enhances the system's predictive capabilities for efficient water usage and

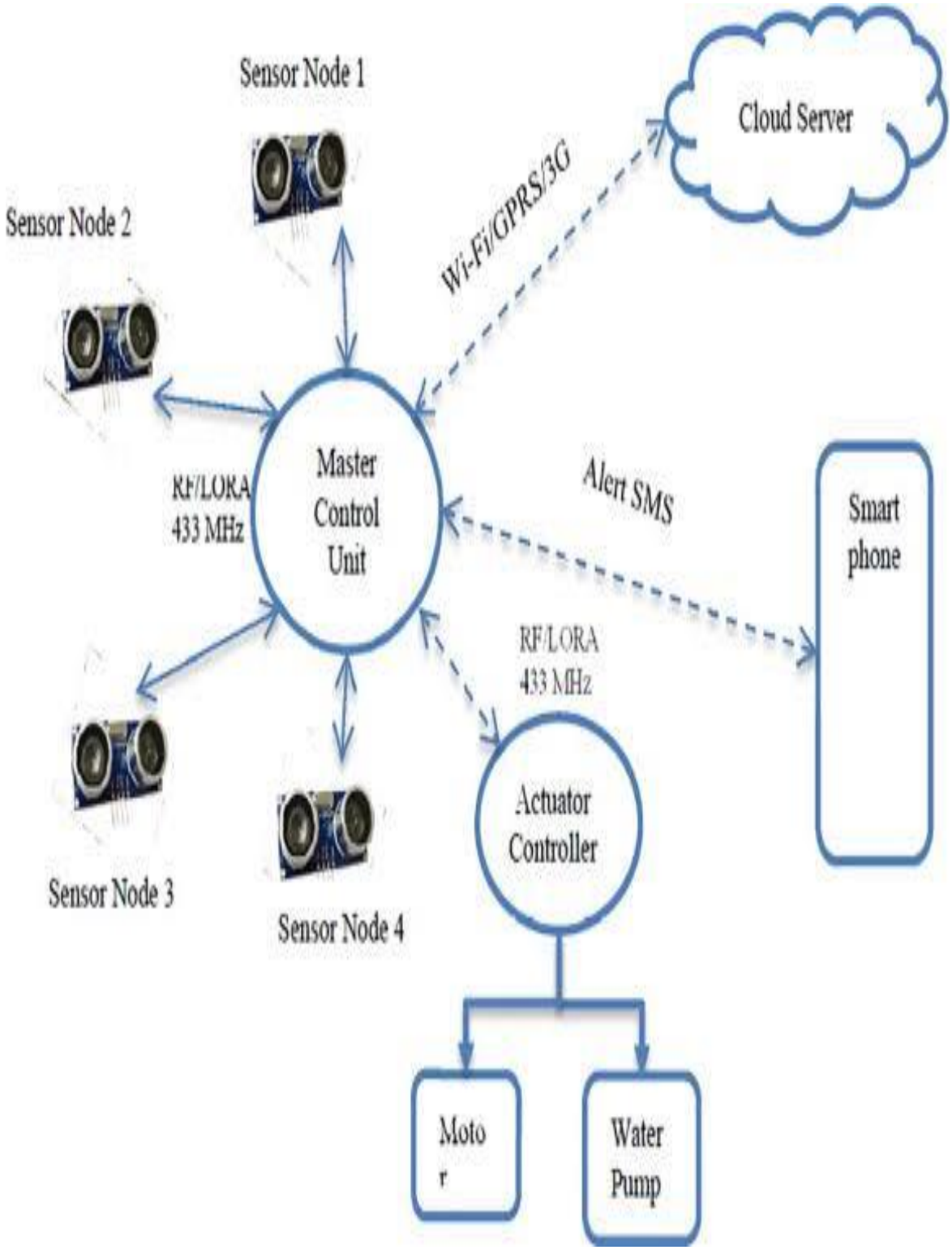
predictive maintenance. We done an simulation of our project in wokwi simulator. The code which we are used is the python script. The primary objectives include ensuring seamless real-time data transmission, enabling remote motor control, and leveraging machine learning to optimize water management processes.

Keyword: Water Monitoring and management system, IoT, Machine learning algorithm.

SMART WATER SYSTEM:



BLOCK DIAGRAM:



"Water System Management Using IoT". Our project aims to use IoT technology to monitor and manage water levels in tanks. The project uses an Arduino microcontroller and a variety of sensors, including water level sensors, temperature sensors, and pressure sensors. The sensors collect data about the water tanks and send it to the Arduino microcontroller. The microcontroller then processes the data and sends it to the cloud using a Wi-Fi module. The cloud server stores the data and provides access to it through a web application. The web application allows users to view the current water levels in their tanks and turn the water pumps on and off. The project can be used to improve water management efficiency and reduce water waste.

For example, the project can be used to automatically turn on the water pump when a tank is low on water and turn it off when the tank is full. This can help to ensure that there is always enough water available, even when there is high demand. The project is also useful for monitoring water quality. The temperature and pressure sensors can be used to detect changes in the water quality, such as the presence of contaminants. This information can be used to alert users to potential problems and take corrective action.

Overall, the project is a well-designed and innovative solution for water management. It uses IoT technology to provide users with real-time information about their water tanks and to automate the water pumping process. This can help to improve water management efficiency, reduce water waste, and protect water quality.

SMART WATER MANAGEMENT BY MACHINE LEARNING ALGORITHM

DATA ANALYSIS AND SYATEM WORKING (Machine Learning):

The data centre receives data from different sensors placed at different locations. The aggregation of the received values will help in decision-making

Tav = TL1+TL2+TL3+…+TLn

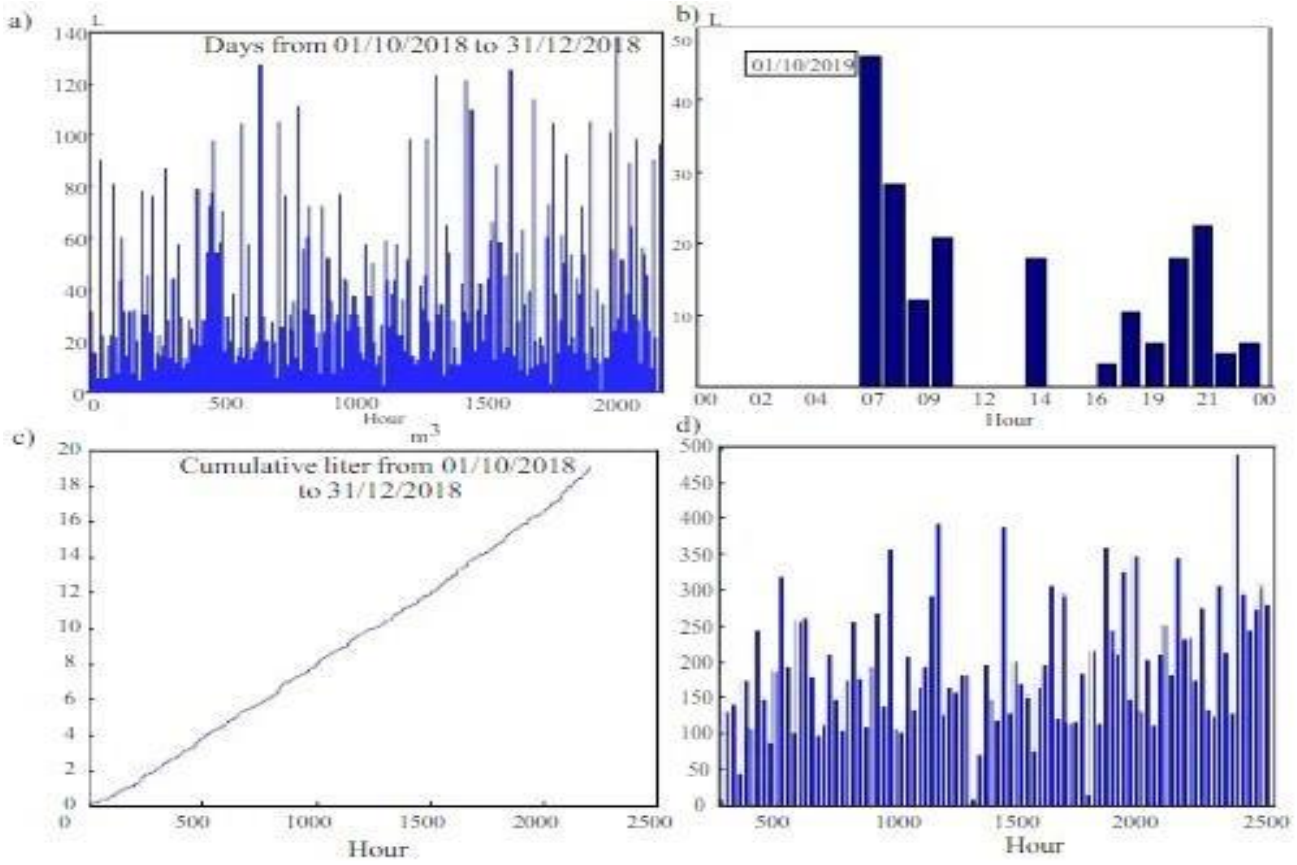
Equation (1) shows aggregation of different temperature values received from temperature sensors at varied locations.Represents TL1 is the location 1, while Tav is the aggregated value of all the sensors along the water supply backbone. Similar equations will be used to calculate aggregated values from all sensors. Specific thresholds will be set for decision-making.

The data centre used machine learning algorithms for decision-making. The algorithms become more accurate as more data is received at the data centre. Since the sensors continuously send data to the data centre, machine learning algorithms build models that can detect any outliers.

Outliers refer to data that falls out of the general or expected data behaviour Such outliers do not only refer to “bad” values but also missing data, meaning the systems can detect when some sensor nodes do not send data to the data centre within the expected intervals. This allows for timely detection

of failed sensors, either by losing connection or battery running. Such occurrences are common in wireless sensor networks. Whenever such outliers are detected in real-time, push notifications are sent to mobile users while appropriate triggers are sent to the municipal or government officials.

Such notifications could be triggered due to sudden change in the water flow rate.

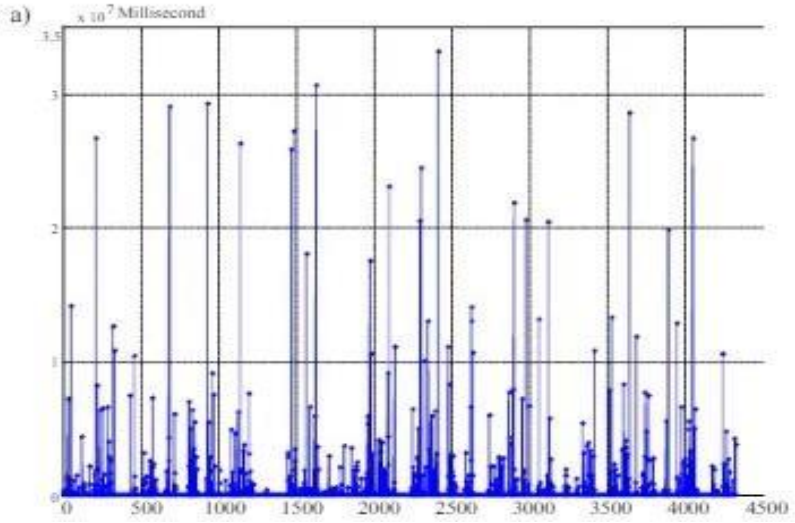


The above graphical representation gives the Water consumption time series:

a) LC from 1 October to 31 December 2018, b) close-up view of the same time series for the first 24hours. c) cumulative water LC over the whole period,d) number of liters consumed per day. Two ML approaches have been implemented for a one-hour water consumption forecasting, the LSTM and the BPNN of the forecast approaches. Here we used a dataset of 4321 events dated in milliseconds,each representing the time difference between two consecutive liters, to predict the next liters of water to be consumed. We

used two learning approaches: LSTM and BPNN. The LSTM model achieved an error (test RMSE) of 13 ms, while the BPNN model achieved an error of 48 ms.

	LSTM	BPNN
Hidden Layer number	2	1
Number of neurons	200/120	150
Activation function	relu/relu	relu
Train RMSE (10^6 ms)	0.33	0.39
Test RMSE (10^6 ms)	0.13	0.48
Total execution time (s)	37.73	24.71



results of two learning approaches are provided in Table. The instant of the next consumed liter of water is predicted respectively with an error (test RMSE) of 13ms and 48ms respectively with the LSTM and the BPNN.

SMART WATER MANAGEMENT STIMULATION PART

CODING:

```
import machine
import time

# Pin assignments for the ultrasonic sensor
TRIGGER_PIN = 23  # GPIO23 for trigger
ECHO_PIN = 22     # GPIO22 for echo

# Pin assignment for the LED
LEAK_LED_PIN = 19  # GPIO19 for the LED

# Set the pin modes
trigger = machine.Pin(TRIGGER_PIN, machine.Pin.OUT)
echo = machine.Pin(ECHO_PIN, machine.Pin.IN)
leak_led = machine.Pin(LEAK_LED_PIN, machine.Pin.OUT)

# Function to measure distance using the ultrasonic sensor
def measure_distance():
    # Generate a short trigger pulse
    trigger.value(0)
    time.sleep_us(5)
    trigger.value(1)
    time.sleep_us(10)
    trigger.value(0)

    # Measure the echo pulse duration to calculate distance
    pulse_start = pulse_end = 0
    while echo.value() == 0:
        pulse_start = time.ticks_us()
    while echo.value() == 1:
        pulse_end = time.ticks_us()

    pulse_duration = pulse_end - pulse_start

    # Calculate distance in centimeters (assuming the speed of sound is 343
m/s)
    distance = (pulse_duration * 0.0343) / 2  # Divide by 2 for one-way travel

    return distance

# Function to check for a water leak
def check_for_leak():
    # Measure the distance from the ultrasonic sensor
    distance = measure_distance()

    # Set the threshold distance for detecting a leak (adjust as needed)
    threshold_distance = 10  # Adjust this value based on your tank setup
```

```
    if distance < threshold_distance:
        # If the distance is less than the threshold, a leak is detected
        return True
    else:
        return False

# Main loop
while True:
    if check_for_leak():
        # Blink the LED to indicate a leak
        leak_led.value(1)  # LED ON
        time.sleep(0.5)
        leak_led.value(0)  # LED OFF
        time.sleep(0.5)
    else:
        leak_led.value(0)  # LED OFF

    time.sleep(1)  # Delay between measurements
```

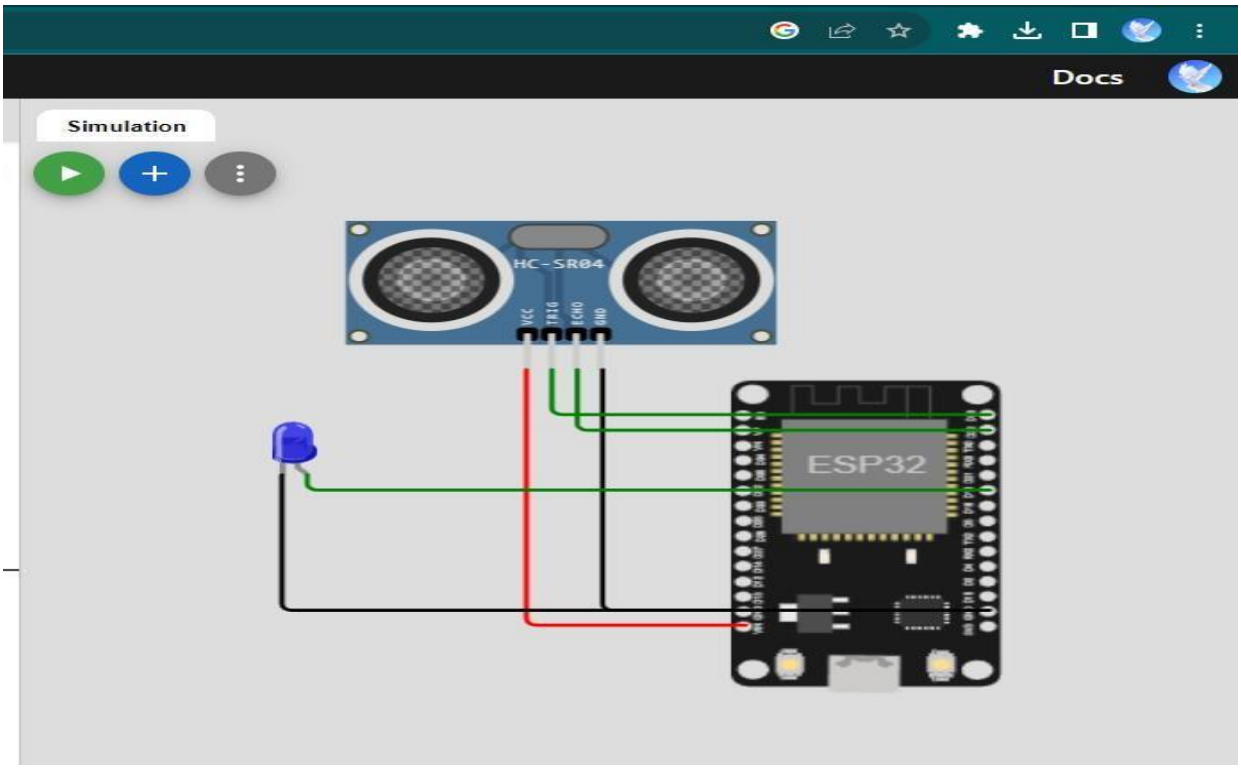
CODING: CODE EXPLANATION:

This code is a Micro Python script designed to detect water leaks using an ultrasonic sensor and indicate the presence of a leak with an LED.Here, the code sets up the pins defined in the previous step, configuring trigger and leak led as output pins and echo as an input pin.

Measure distance Function: This function is responsible for measuring the distance using the ultrasonic sensor. It triggers the sensor, measures the echo pulse duration, and calculates the distance based on the speed of sound.The distance in centi meters is returned.Check for leak Function: This function checks if a water leak is detected by calling measure distance. It compares the measured distance with a threshold distance, and if the measured distance is less than the threshold, it returns True to indicate a leak. Otherwise, it returns False.The main loop continuously checks for leaks

using the check_for_leak function. If a leak is detected, it blinks the LED for visual indication. The LED is turned on for 0.5 seconds and then turned off for 0.5 seconds. If no leak is detected, the LED remains off.

SIMULATION CIRCUIT:



SIMULATION HYPERLINK:

TO SEE OUR OUTPUT YOU CAN VISIT OUR LINK

https://wokwi.com/projects/379568578162000897

SMART WATER SYSTEM MOBILE APP INTERFACE

APP DEVELOPMENT:

We created an app for our project SMART WATER MANAGEMENT. Which it monitors the level of the water surface in the tank. If the threshold level of water surface decrease then our project will make the motor to turn ON by using our app. It can be accessed by the remote user anytime and anywhere.

PYTHON CODE FOR IMPLEMENTATION IN FIREBASE:

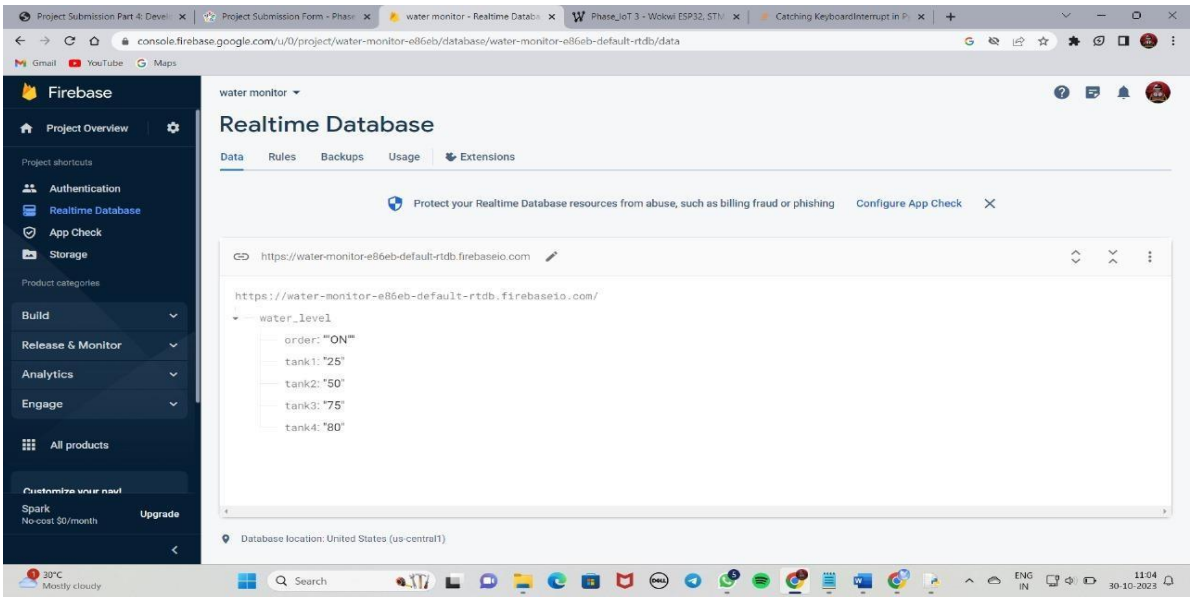
```
import firebase_admin
import sys
import time
from firebase_admin import credentials,db

c=credentials.Certificate('C:/Users/havoc/Downloads/water-monitor- e86eb-firebase-adminsdk-1ww9s-31fd2fd2f0.json')

firebase_admin.initialize_app(c,{'databaseURL':'https://water-monitor-e86eb-default-rtdb.firebaseio.com/'})
L=db.reference('water_level/tank1')
L.set('25')
```

CODE EXPLAINATION:

The code we provided is a Python script that uses the Firebase Admin SDK to write data to a Firebase Realtime Database. The code first imports the necessary libraries, including firebase_admin, sys, time, and db. Then, it creates a credential object using the certificate file that you downloaded from Firebase. Next, it initializes the Firebase Admin SDK with the credential object and the database URL. The code then creates a reference to the water_level/tank1 node in the database. Finally, it uses the set() method to write the value "25" to the node.

REALTIME WATER LEVEL IN TANK:

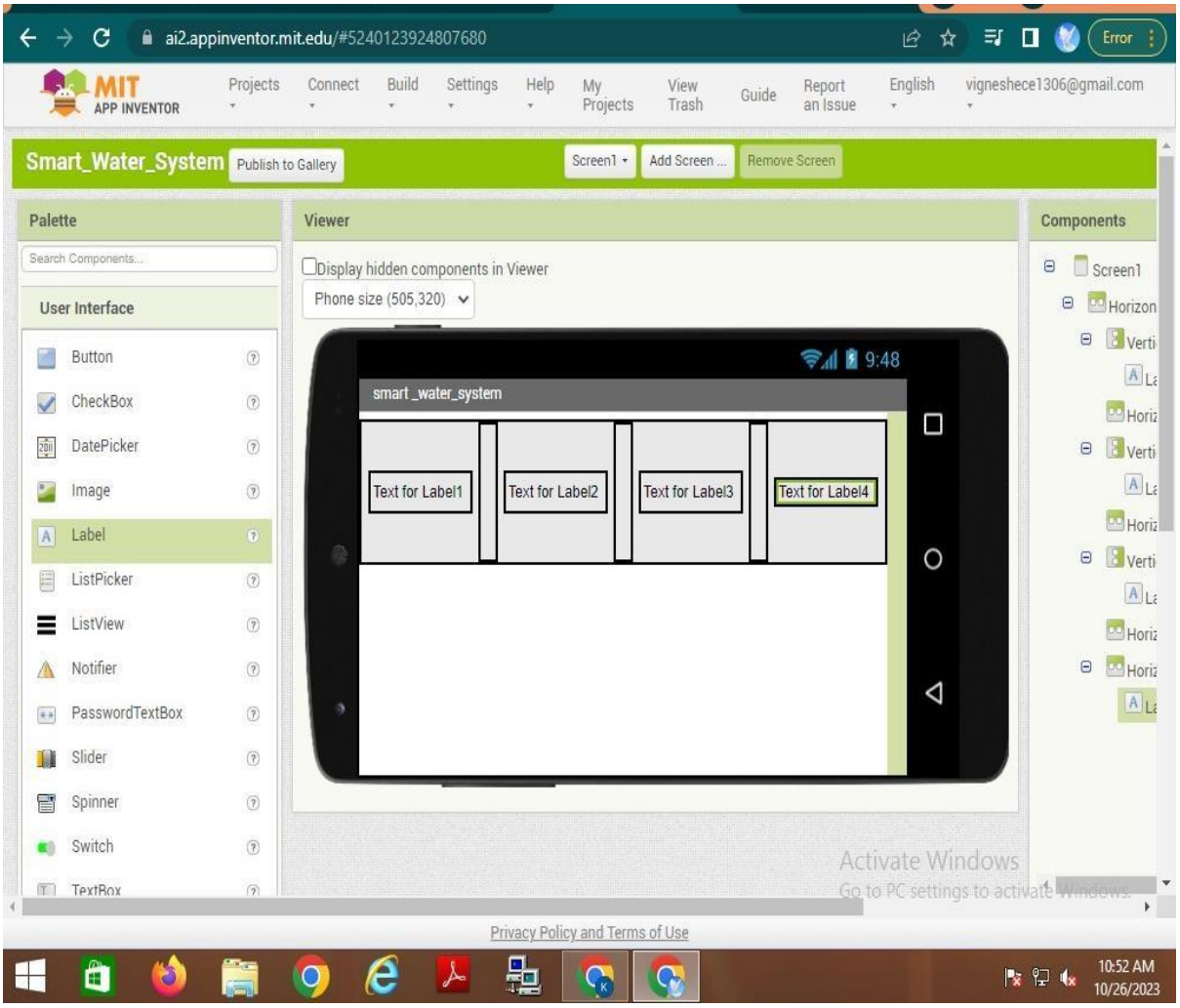
Here we donate the level of the water level in tank. We donated the levels like 25 for tank 1, 50 for tank 2, 75 for tank 3, 80 for tank 4. We indicate the levels in Firebase website. The following Picture will describes our work in the firebase application.



WEBSITE DEVELOPMENT:

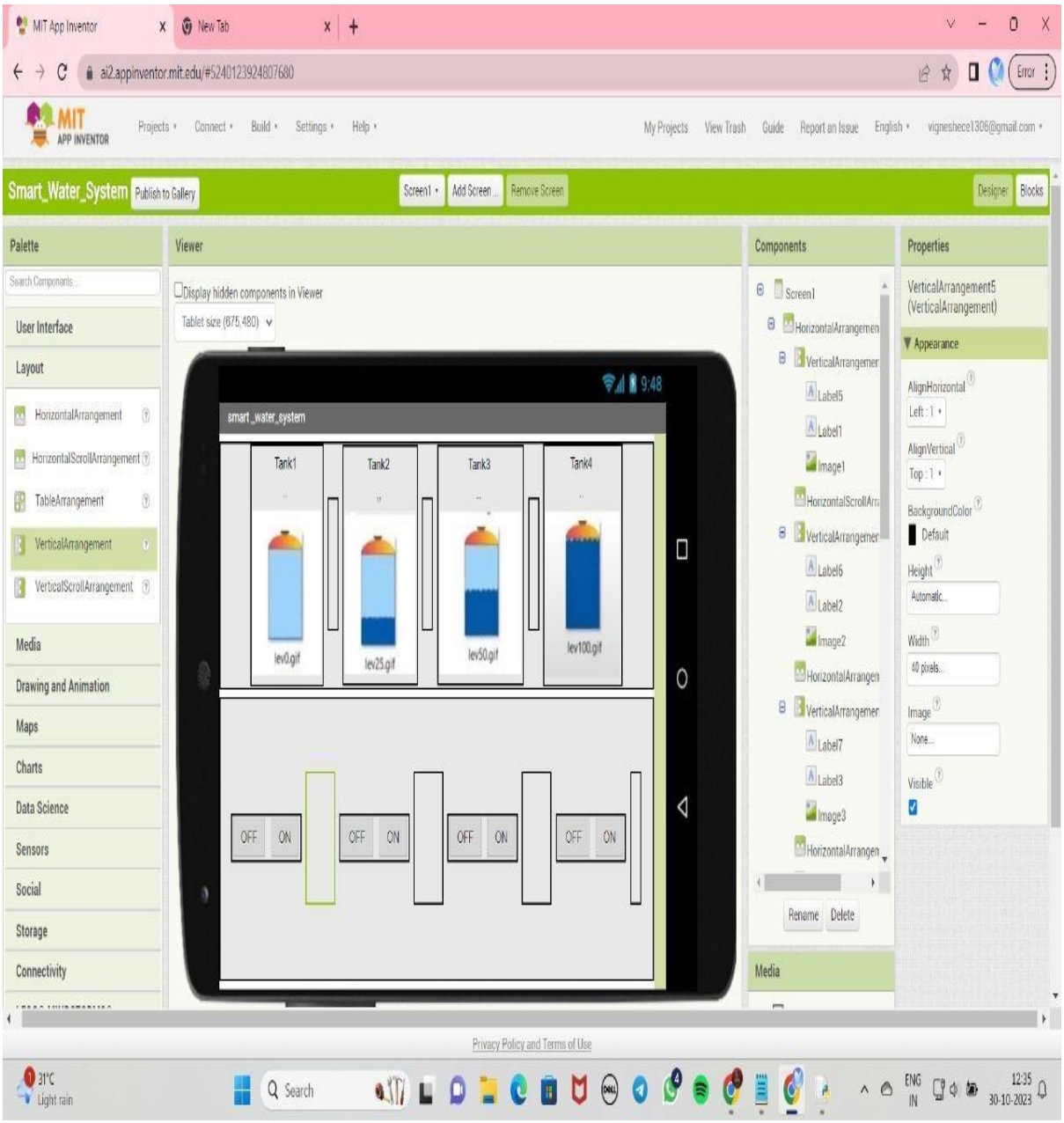
As we mentioned above, we created a website for our project “SMART WATER SYSTEM”. The following pictures will describe our work to create the website.

STEP 1:



Designing our website. In the above picture we split the screen into 4 because we are using 4 tanks to measure the water level. After that we paste our Firebase URL and firebase token.

STEP2:

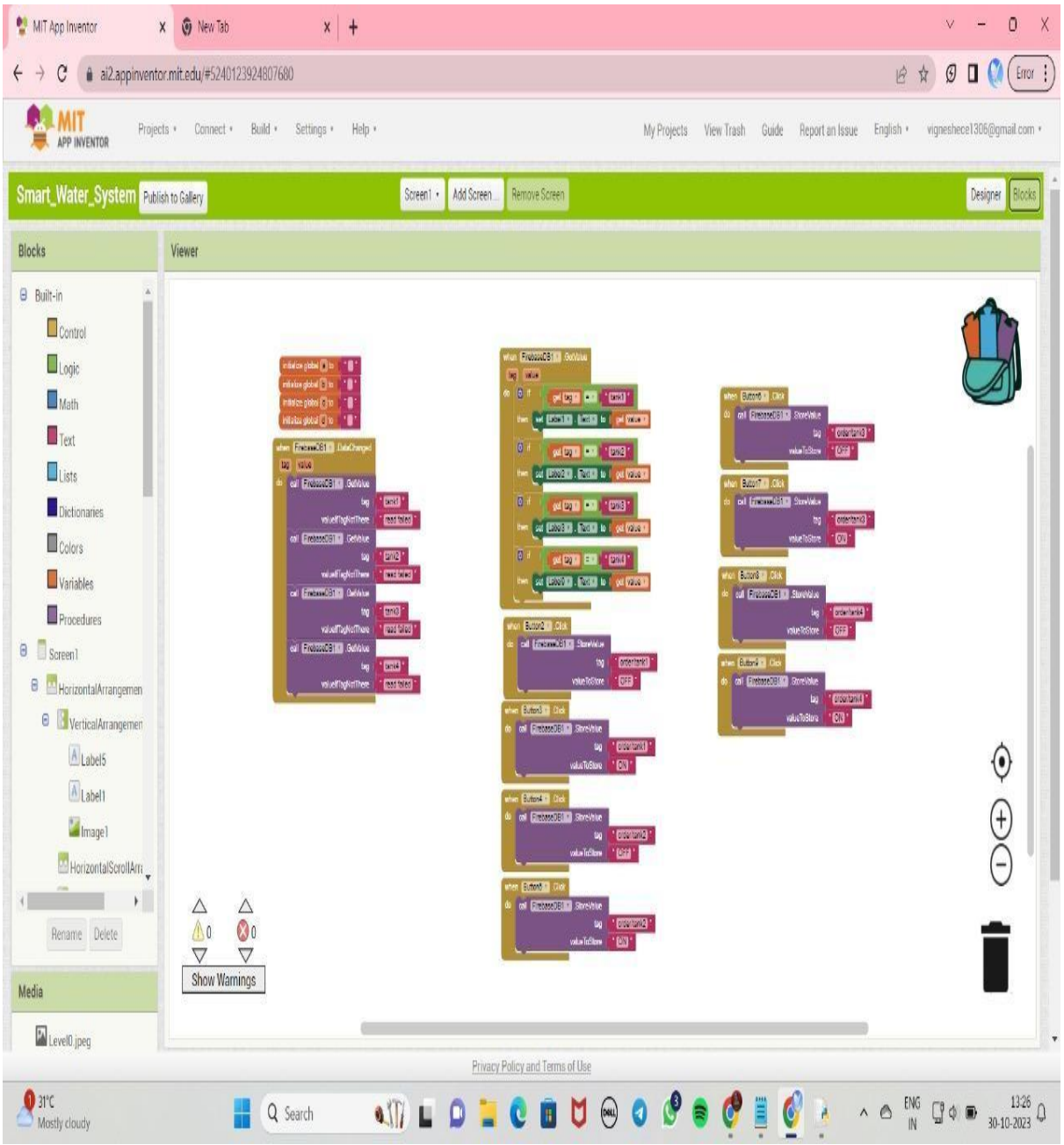


After that we uploading our image in that 4 columns. The following picture will describe the water level in the tank. The ON and OFF button describes the motor function, Which is accessible for the remote user in our application.

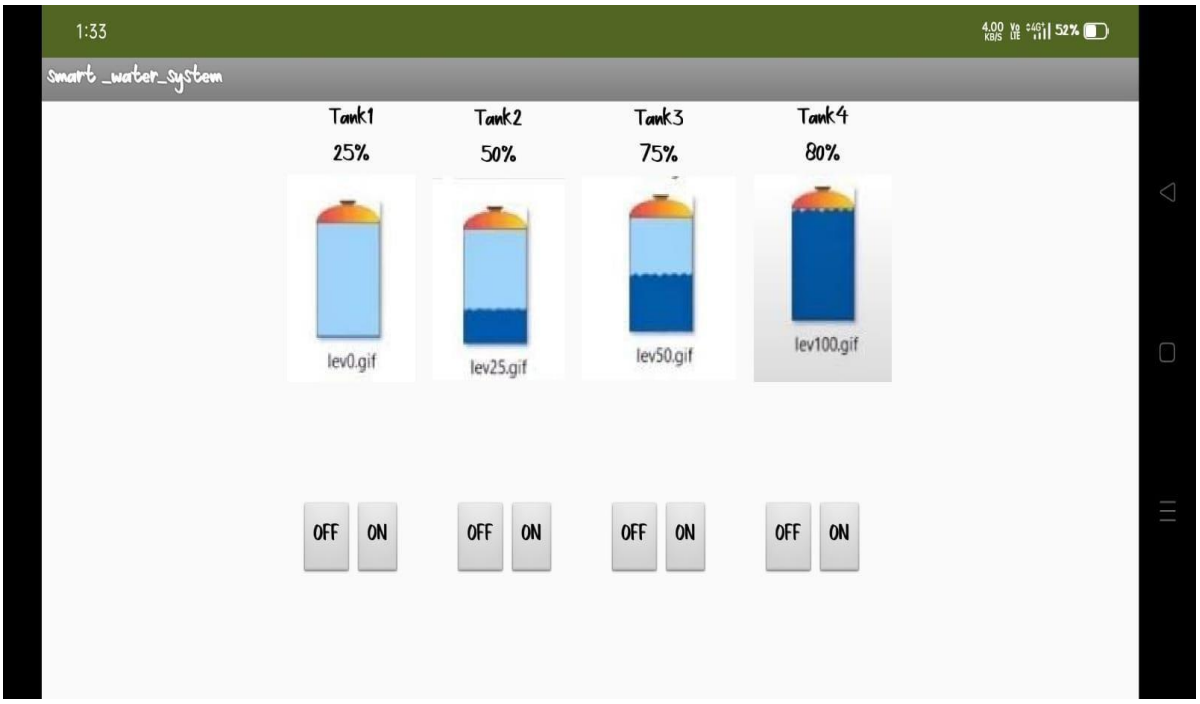
STEP 3:

Then we went to the block section and completing the code for creating an application to be accessed by the remote user.

We are building the code for 4 tanks consisting of different water levels in each tank.



USER INTERFACE:



- ✓ The above picture shows a screenshot of your project took by the mobile user, "Water System Management Using IoT".

CONCLUSION:

The "Water System Management Using IoT" project is a successful demonstration of how IoT technology can be used to improve water management efficiency and reduce water waste. The project uses a variety of sensors and an Arduino microcontroller to collect data about water levels, temperature, and pressure in tanks. This data is then sent to the cloud using a Wi-Fi module and stored on a cloud server. A web application allows users to view the data and control the water pumps.

”KEEP YOUR WATER TANK ON TAP WITH OUR APP”

HYPERLINK:

TO VISIT OUR APP HERE WE PROVIDE OUR RESPECTIVE URL FOR BOTH APP/WEBSITE AND SIMULATION WE DONE IN WOKWI

https://ai2.appinventor.mit.edu/#5240123924807680