SMART WATER MANAGEMENT

AIM:

To build 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.

Micropython code to get the data from ESP32 and its IOT sensors:

```
from machine import Pin
import time
import network
import urequests

WIFI_SSID = 'Wowki-GUEST'
WIFI_PASSWORD = ''
SERVER_IP = 'http://localhost:8080'
SERVER_PORT = '8080'

# Connect to WiFi
wifi = network.WLAN(network.STA_IF)
wifi.active(True)
wifi.connect(WIFI_SSID, WIFI_PASSWORD)

# Wait for the connection to be established
while not wifi.isconnected():
    time.sleep(1)

# Define the pins for your sensors
FLOW_SENSOR_PIN = 14
PRESSURE_SENSOR_PIN = 27
LEAK_SENSOR_PIN = 32

# Set up the pins
flow_sensor = Pin(FLOW_SENSOR_PIN, Pin.IN)
pressure_sensor = Pin(PRESSURE_SENSOR_PIN, Pin.IN)
leak_sensor = Pin(LEAK_SENSOR_PIN, Pin.IN)

def read_sensors():
    flow_rate = flow_sensor.value()
    pressure = pressure_sensor.value()
    leak_detected = leak_sensor.value()
```

```
    # Send the sensor values to the server
    url = 'http://{}:{}/update'.format(SERVER_IP, SERVER_PORT)
    headers = {'content-type': 'application/json'}
    data = {'flow_rate': flow_rate, 'pressure': pressure, 'leak_detected': leak_detected}
    response = urequests.post(url, json=data, headers=headers)

while True:
    read_sensors()
    time.sleep(1)
```

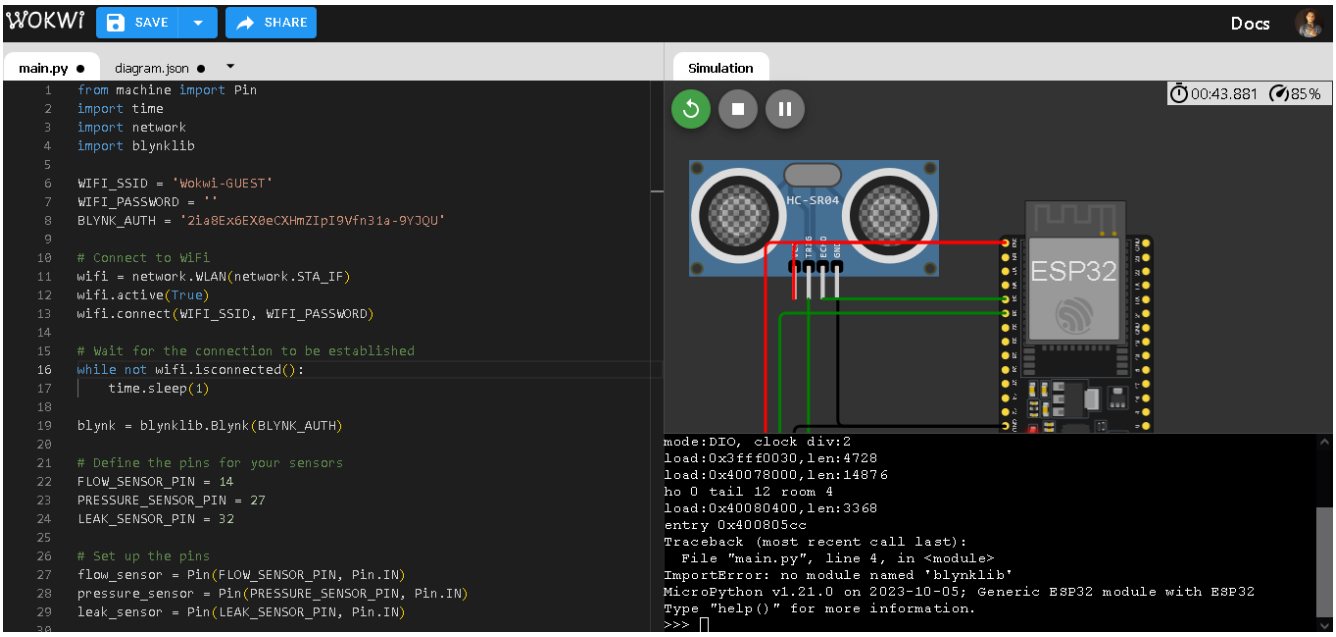
To create a HTTP server using Node.js

```
var http = require('http');
var fs = require('fs');

http.createServer(function (req, res) {
    fs.readFile('yourfile.html', function(err, data) {
        res.writeHead(200, {'Content-Type': 'text/html'});
        res.write(data);
        return res.end();
    });
}).listen(8080);
```

HTML Code to view real time data sharing:

```
<!DOCTYPE html>
<html>
<head>
    <title>Real Time Data</title>
    <script>
        var websocket;
        window.addEventListener('load', onLoad);

        function onLoad(event) {
            websocket = new WebSocket('ws://your-esp32-ip-address');
            websocket.onopen = onOpen;
            websocket.onclose = onClose;
            websocket.onmessage = onMessage;
        }

        function onOpen(event) {
            console.log('Connection opened');
        }

        function onClose(event) {
            console.log('Connection closed');
            setTimeout(() => onLoad(null), 5000);
```

```
        }

        function onMessage(event) {
            var data = event.data;
            document.getElementById('data').innerHTML = data;
        }
    </script>
</head>
<body>
    <h1>Real Time Data from ESP32</h1>
    <p id="data"></p>
</body>
</html>
```



Conclusion:

Using micropython code in wokwi to get the realtime data from ESP32 and IOT sensors and by creating HTTP server using Node.js to write a HTML code to visualize and store the data we got from the IOT sensors has been performed above.