SMART WATER MANAGEMENT
PHASE 2

INNOVATION:

Integration with IOT Devices:

ESP32 Development Board: Start with an ESP32 development board, which provides Wi-Fi capabilities and GPIO pins for connecting sensors.

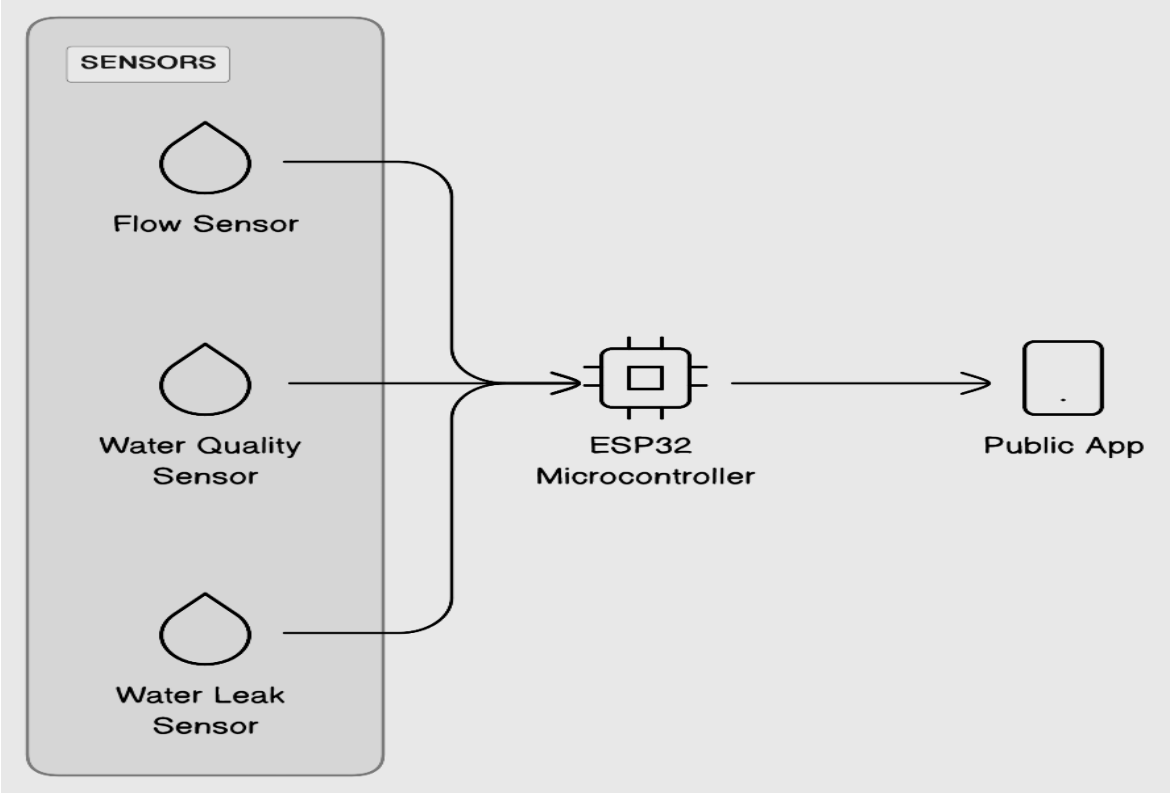
Water Flow Sensor: Connect the water flow sensor to the ESP32 to measure the water flow rate. These sensors typically have a digital output that can be read by the ESP32.

Water Leak Sensor: Connect the water leak sensor to the ESP32. These sensors often have digital outputs that can detect the presence of water.

Water Pressure Sensor: Connect the water pressure sensor to the ESP32. Depending on the sensor, it might communicate using I2C, SPI, or analog signals.

Water Quality Sensor: Connect the water quality sensor to the ESP32. These sensors may use various interfaces, such as I2C, UART, or analog signals, depending on the specific sensor model.

Block Diagram:



Data Sharing Platforms:

LoRaWAN: LoRaWAN (Long Range Wide Area Network) is a low-power, long-range wireless communication technology ideal for IoT devices. IoT sensors built with LoRaWAN capability can transmit data to gateways, which forward it to the data sharing platform via backhaul connections like Ethernet or cellular networks.

PLATFORM:

Using available open source/cloud based platforms to analyse, collect and store data can reduce workload and cost.

- SmartWater is a service-oriented and sensor cloud-based framework for smart monitoring of water environments . It combines cutting-edge technologies in the field of sensor clouds, deep learning, knowledge reasoning, and data processing and analytics
- Arduino IDE Software is used to program the ESP32 board

ALGORITHM:

1. **Data Collection**: Collect data from IoT devices such as temperature, humidity, and pressure sensors using LoRaWAN.
2. **Data Processing**: Process the collected data to identify patterns and trends in the data.
3. **Data Transmission**: Transmit the processed data from the IoT devices to the LoRaWAN network server using a LoRa transceiver module.
4. **Data Reception**: Receive the transmitted data at a LoRaWAN gateway connected to the network server.
5. **Data Storage**: Store the received data in a database or other storage medium for further analysis.
6. **Web Application Development**: From a web application that uses HTTP to communicate with the LoRaWAN network server and retrieve the stored data.
7. **Data Visualization**: Visualize the retrieved data in real-time using a SmartWater Platform.
8. **Machine Learning**: Train a machine learning model using the processed data to predict water consumption patterns and identify areas where water conservation can be improved.
9. **Conservation Suggestions**: Use the machine learning model to provide conservation suggestions based on the predicted water consumption patterns.