FACULTY OF ENGINEERING AND TECHNOLOGY

JAIN DEEMED-TO-BE UNIVERSITY

# PROJECT REPORT

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND SYSTEMS ENGINEERING
(INTERNET OF THINGS)
August 2024

—

# Smart Water Management Technology (SWMT)

—


Submitted by

SHAHD K
Name: HERVE C. INEZA USN: 21BTRCO044
Name: RAVI GIRI USN: 21BTRCO045
Name: AMARA A. KAMARA USN: 21BTRCO048
Name: DJATASSIBA YAOVI JOËL USN: 21BTRCO055

Under the guidance of

Dr. SURESH KALLAM

Head Of Departement

Faculty of Engineering & Technology

Jain (Deemed-To-Be University)

Department of Computer Science & Engineering

Jain Global Campus, KanakapuraTaluk - 562112

Ramanagara District, Karnataka, India

# INTRODUCTION

Water is the cornerstone of life on Earth, playing an indispensable role in ecosystems, agriculture, industry, and daily human activities. Despite its fundamental importance, water is becoming an increasingly scarce and contested resource. The pressures on global water resources are mounting due to several interrelated factors, including rapid urbanization, industrial growth, population expansion, and the escalating impacts of climate change. These dynamics are altering weather patterns, intensifying droughts, and leading to more frequent and severe water shortages. As a result, managing water efficiently has become one of the most pressing challenges of our time.

Traditional water management systems, while effective in their time, are struggling to keep pace with these new realities. They often rely on outdated infrastructure and manual processes that are unable to provide the real-time data or automated responses necessary to manage water resources effectively in today's complex environment. In many parts of the world, water management is still characterized by inefficiencies such as delayed responses to leaks, over-reliance on human monitoring, and a lack of integration between different components of the water system. These shortcomings contribute to significant water wastage, higher operational costs, and, in some cases, the contamination of water supplies.

This project, centered around the development of a Smart Water Management Technology (SWMT) system, aims to harness these capabilities to create a more responsive and intelligent water management solution. The SWMT system is designed to optimize water usage, monitor water quality, and provide users with actionable recommendations based on real-time data and predictive insights. By doing so, it not only addresses the immediate needs of water management but also contributes to the broader goal of sustainable water use in the face of global challenges.

# METHODOLOGY

The development of the Smart Water Management Technology (SWMT) system involved a comprehensive approach that combined hardware and software components, iterative testing, and a user-centered design process. The methodology for the SWMT project can be broken down into several key phases: literature review, system design, hardware implementation, software development, data collection and analysis, testing, and validation.

## A. Literature Review

The project began with an extensive literature review to understand the current state of water management technologies and identify gaps that the SWMT system could address. This review included research on IoT-based water monitoring systems, water quality sensors, automated water distribution systems, and AI-driven predictive analytics. The insights gained from this review informed the design and functionality of the SWMT system, ensuring that it incorporated the latest advancements in technology while addressing existing limitations.

## B. System Design

The system design phase involved defining the architecture and components of the SWMT system. The system was conceptualized as a modular solution comprising several key components:

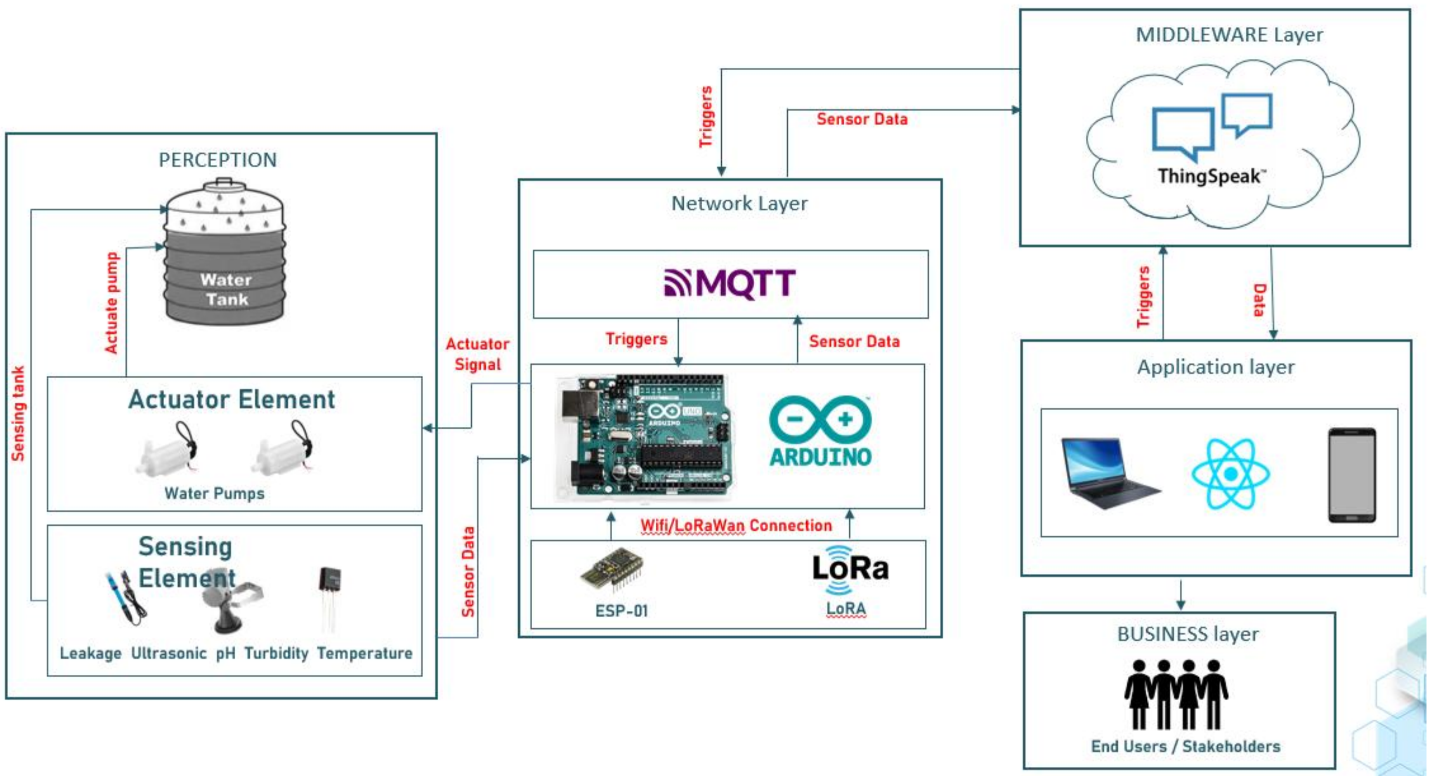
**Perception layer**: A network of sensors was designed to monitor various parameters such as water level, quality, and temperature. Sensors were selected based on their accuracy, reliability, and compatibility with the microcontroller.

**Network layer**: An Arduino-based microcontroller was chosen to interface with the sensors and process the data. The ESP32 Wi-Fi module and LoRa module was integrated into the system to enable wireless communication between the sensors and the cloud platform.

**Middleware Layer**: The cloud platform was designed to store and analyze data collected from the sensors. This platform supports real-time monitoring, data visualization, and predictive analytics, and it can be accessed via a web or mobile application.

**Application layer**: A user-friendly interface was designed. This interface allows users to monitor system performance, view data, and control the system remotely. It was developed for both web and mobile platforms for accessibility.

**Business layer**: This layer houses the core decision-making rules. It processes data from the sensors and predictive models to trigger automated actions, such as pump activation for refilling tanks and sending alerts for leak detection.



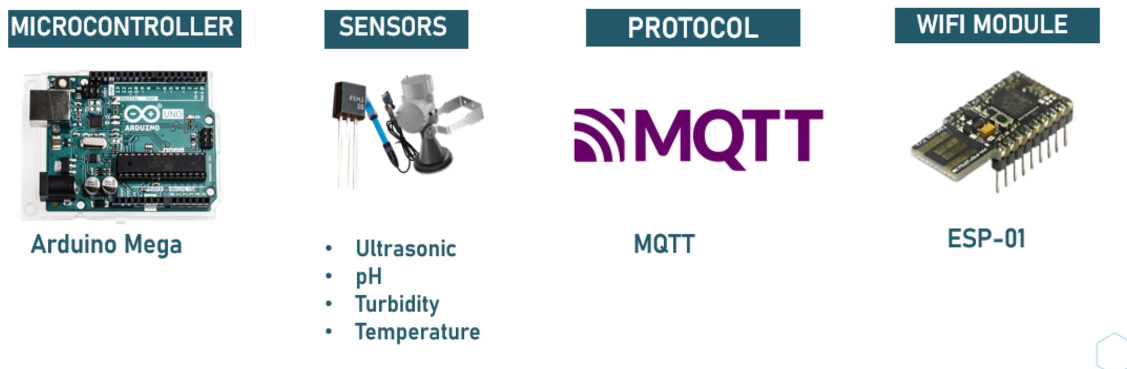

## C. Hardware Implementation

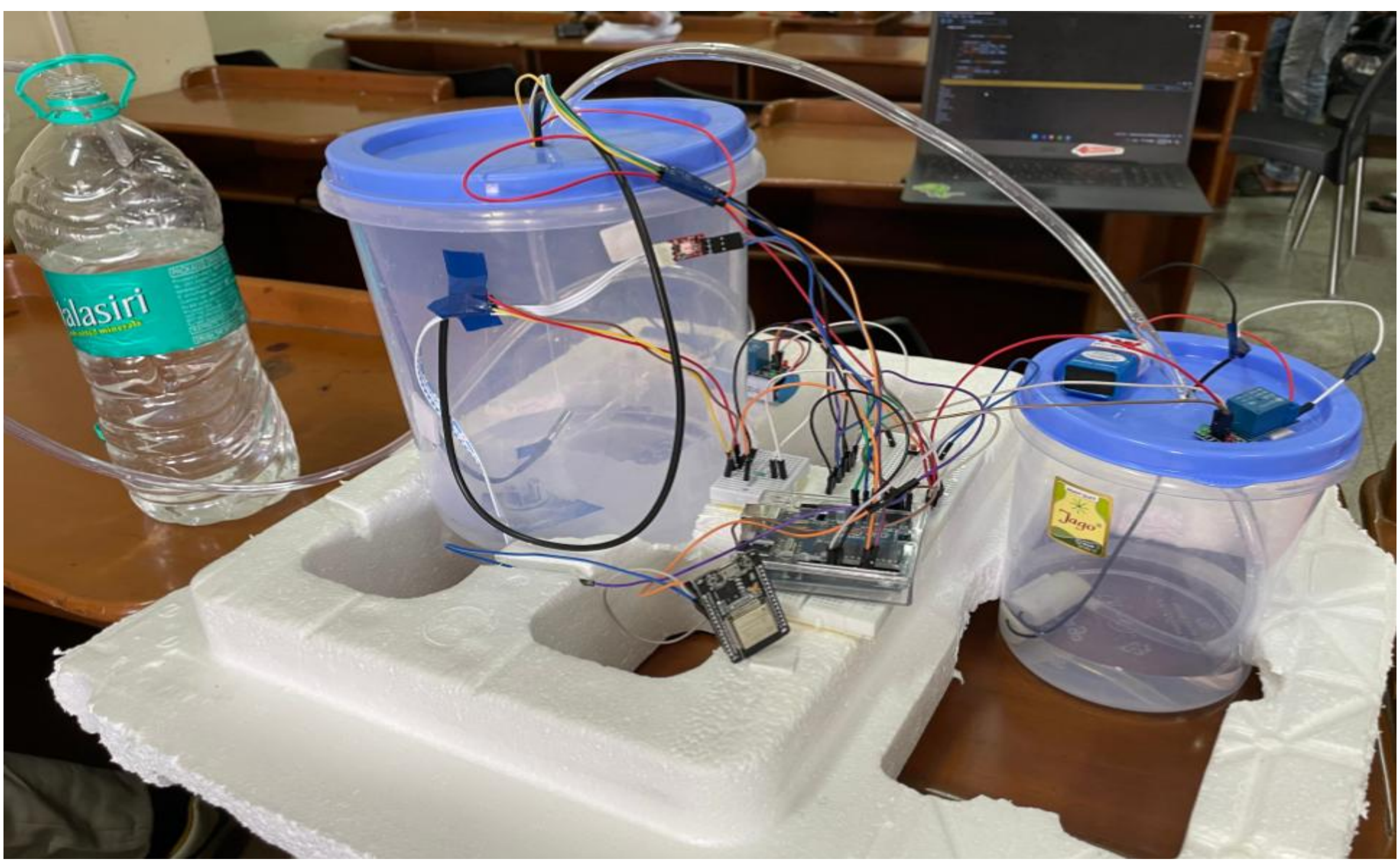
The hardware implementation phase involved the assembly and configuration of the physical components of the SWMT system. The key steps included:

**Sensor Integration**: Sensors were carefully selected, calibrated, and integrated with the Arduino microcontroller. The water level sensor, turbidity sensor, pH sensor, and temperature sensor were connected to the microcontroller, which processed the sensor data in real time.

**Power Management**: Power management strategies were implemented to ensure the system's energy efficiency. This included selecting low-power components and optimizing the microcontroller’s power consumption.

**Prototype Development**: A working prototype of the SWMT system was developed, combining all hardware components into a single, cohesive unit. The prototype was housed in a waterproof enclosure to protect the electronics and ensure durability in various environmental conditions.



## D. Software Development

The software development phase involved creating the code that would run on the microcontroller, as well as developing the cloud platform and user interface. The key components of the software included:

**Microcontroller Programming**: The Arduino was programmed to collect data from the sensors, process it, and send it to the cloud platform via the ESP32 module. The code also included logic for controlling the water pump and valve based on predefined conditions and user inputs.

**Cloud Platform Development**: The cloud platform was built using a combination of cloud services, databases, and analytics tools. The platform was designed to store large volumes of data, perform real-time analysis, and generate predictive insights. It also included features for data visualization and alert generation.

**User Interface Design**: The user interface was developed to provide an intuitive experience for users. The design focused on ease of use, clear data presentation, and responsiveness across devices. The interface allowed users to view real-time data, access historical data, and receive notifications about system performance and potential issues.

MOBILE APP

React App

CLOUD SERVICE

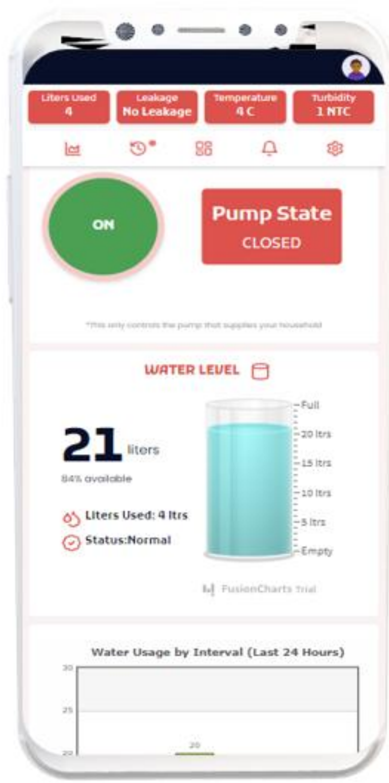
ThingSpeak

ThingSpeak

Embedded Programming

ARDUINO

Arduino



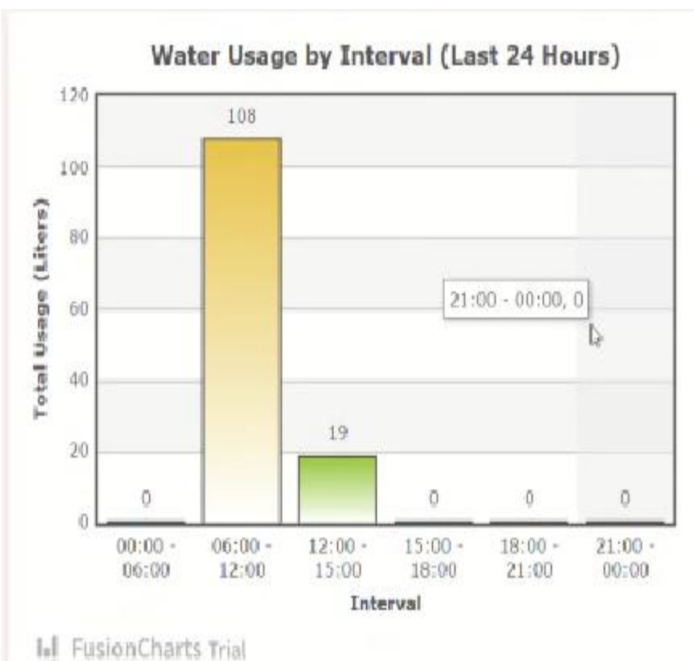
## E. Data Collection and Analysis

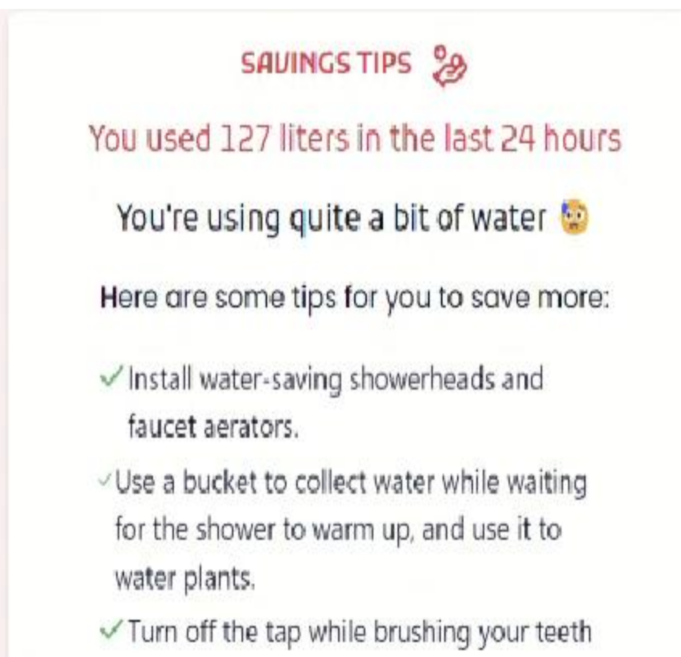
Data collection and analysis were critical components of the SWMT project. During this phase:

**Data Acquisition**: The system collected data on water levels, quality, temperature, valve activations, and other relevant parameters over an extended period. The data was transmitted to the cloud platform for storage and analysis.

**Data Processing**: The raw data was processed to remove noise and outliers. This involved filtering techniques and normalization to ensure the data was clean and ready for analysis.

**Predictive Analytics**: Machine learning algorithms were developed and trained on the collected data to predict future water usage patterns, potential water shortages, and optimal times for tank refilling. The algorithms were iteratively refined to improve their accuracy and reliability.



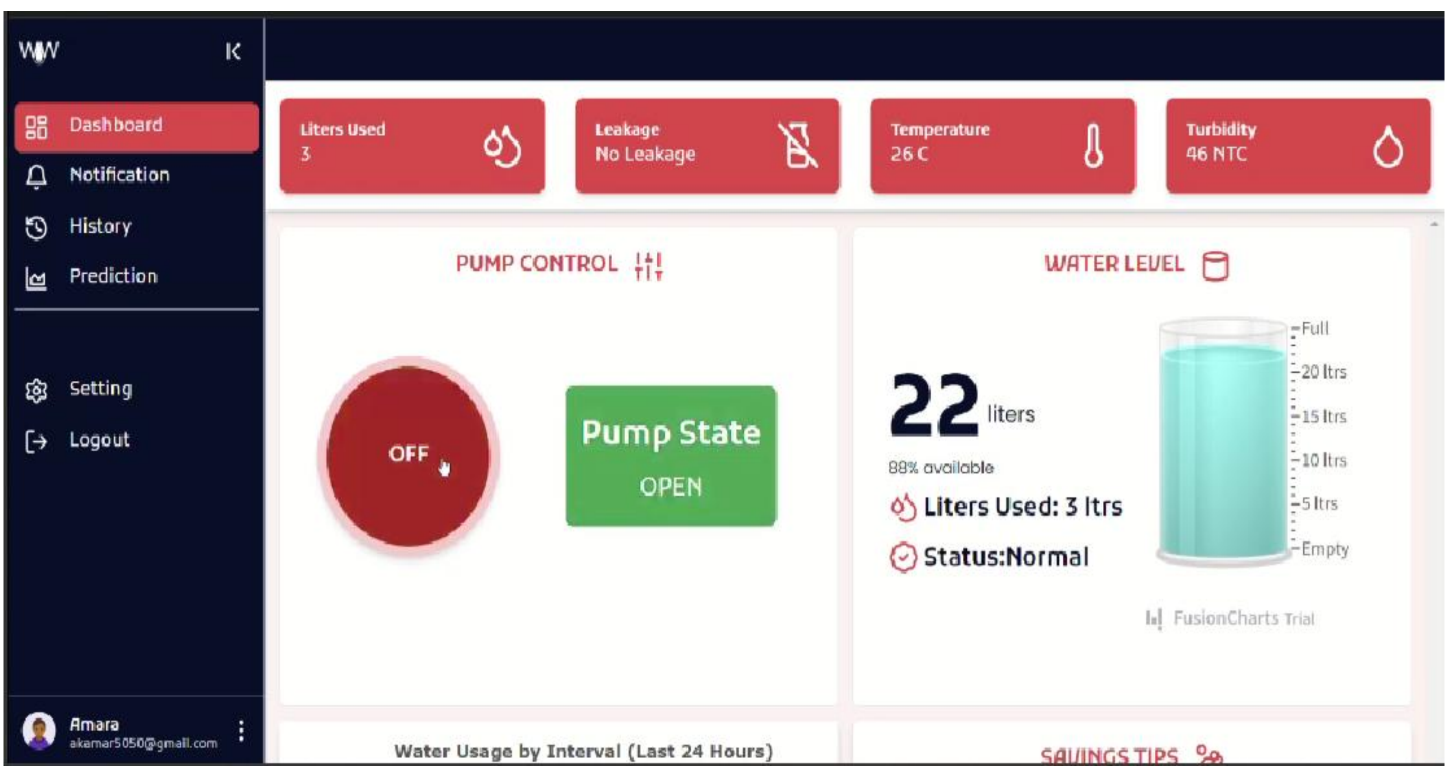
## F. Testing and Validation

The testing and validation phase ensured that the SWMT system met its design objectives and performed reliably under various conditions. This phase included:

**Functional Testing**: Each component of the system was tested to verify its functionality. This included checking sensor accuracy, communication reliability, and the correct operation of the microcontroller and cloud platform.

**System Integration Testing**: The entire system was tested as a whole to ensure that all components worked seamlessly together. This testing involved simulating different scenarios, such as varying water levels, temperature changes, and potential leaks, to verify the system's responsiveness and accuracy.

**Performance Evaluation**: The system's performance was evaluated based on criteria such as data accuracy, response time, energy consumption, and the effectiveness of the predictive analytics. The results of this evaluation informed any final adjustments needed before the system could be deployed in a real-world setting.



# RESULTS

The implementation of the Smart Water Management Technology (SWMT) system yielded significant and promising results, demonstrating the effectiveness of the system in addressing the challenges of water management. The outcomes of the project are summarized below, highlighting the system's performance in real-time monitoring, automated control, water conservation, and user engagement.

### A. Real-Time Monitoring and Data Accuracy

One of the primary objectives of the SWMT system was to provide accurate, real-time monitoring of key water parameters such as water level, quality, and temperature. The system successfully achieved this goal, with the following results:

**Water Level Monitoring**: The water level sensors consistently provided accurate measurements, allowing users to monitor tank levels in real-time. The data collected showed a high degree of precision, with minimal variance between the actual and recorded water levels. This enabled users to make informed decisions about water usage and storage.

**Water Quality Monitoring**: The turbidity sensors effectively measured water quality, providing a percentage-based quality score. The system was able to detect fluctuations in water quality and alert users to potential contamination. This feature proved particularly valuable in ensuring the safety of water supplies, especially in areas where water quality is a concern.

**Temperature Monitoring**: The system's temperature sensors accurately tracked both the current water temperature and the future surrounding temperature (obtained via API). This data allowed the system to predict how temperature changes could affect water usage patterns and storage conditions.

### B. Automated Control and Efficiency

The SWMT system's automated control features were designed to optimize water usage and reduce wastage. The results of the automated control functions include:

**Automated Tank Refilling**: The system successfully automated the refilling of water tanks based on predefined conditions, such as water level thresholds and predicted usage patterns. The automation reduced the need for manual intervention, ensuring that tanks were refilled only, when necessary, thus preventing overflows and conserving water.

**Energy Efficiency**: By automating the pump and valve operations based on real-time data, the system optimized energy usage. The SWMT system's ability to operate pumps and valves only when required resulted in a noticeable reduction in energy consumption, translating into cost savings for users.

### C. Water Conservation and Usage Optimization

One of the key goals of the SWMT system was to promote water conservation by providing users with actionable insights and recommendations. The system's impact on water conservation is evidenced by the following results:

**Reduction in Water Wastage**: The combination of real-time monitoring, predictive analytics, and automated control significantly reduced water wastage. The system's ability to detect and prevent overflows, and inefficient water usage patterns contributed to a more sustainable use of water resources.

**Optimized Water Usage Patterns**: The predictive analytics component of the system successfully identified patterns in water usage based on historical and current data. By analyzing factors such as time of day, water level, and temperature, the system provided users with recommendations on the optimal times to use or conserve water. Users reported that these recommendations helped them manage their water consumption more effectively, leading to a reduction in overall usage.

**Behavioral Change**: The user-friendly interface and real-time alerts encouraged users to be more conscious of their water usage. Over time, users became more proactive in managing their water resources, adjusting their usage habits to align with the system's recommendations. This behavioral change played a crucial role in the success of the water conservation efforts.

### D. User Engagement and Satisfaction

The SWMT system was designed with the user experience in mind, ensuring that it was accessible, informative, and easy to use. The results in terms of user engagement and satisfaction include:

**Increased Awareness**: The system's data visualization tools and informative interface increased users' awareness of their water consumption patterns. This awareness led to more informed decision-making and a greater commitment to water conservation practices.

**Adoption and Scalability**: The successful implementation and positive reception of the SWMT system have paved the way for potential scalability. The system's modular design and adaptability make it suitable for deployment in a variety of settings, from residential homes to larger commercial or industrial facilities.

# Project Implementation Guide

This comprehensive, step-by-step guide will walk you through implementing the Smart Water Management Technology (SWMT) project, from hardware wiring to deploying the web-based dashboard, based on your provided files and architecture.

## Phase 1: Hardware Assembly and Wiring

This phase involves setting up the physical components and establishing serial communication between the two microcontrollers.

1. Mounting and Power: Set up your two water containers (tanks) and mount the Arduino and ESP32 microcontrollers on a stable surface. Ensure both boards and the H-Bridge motor driver for the pumps have a stable power supply and share a common ground reference.

2. Sensing Element Wiring (to Arduino): Connect all sensors to the Arduino as follows:
    - Temperature (DS18B20): Data pin to Arduino Digital Pin 3.
    - Ultrasonic Sensor (Level): Trig to Arduino Digital Pin 2 and Echo to Arduino Digital Pin 4.
    - Turbidity Sensor: Analog output to Arduino Analog Pin A0.

3. Actuator Wiring (to Arduino): Connect the control pins to the H-Bridge driver, which then powers the Water Pumps:
    - Pump 1 (Automated): Connect H-Bridge inputs to Arduino Digital Pins 6 and 7.
    - Pump 2 (Remote/Local): Connect H-Bridge inputs to Arduino Digital Pins 8 and 9.

4. Inter-Processor Serial Communication: Establish the local data link between the two boards using the defined serial pins:
    - Arduino TX (Pin 11) → ESP32 RX (Pin 16)
    - Arduino RX (Pin 10) → ESP32 TX (Pin 17)

### Phase 2: Arduino Programming (Sensing & Actuation)

The Arduino is the control unit, responsible for readings and pump management.

1. IDE Setup and Libraries: Install the required libraries in the Arduino IDE: OneWire, DallasTemperature, SoftwareSerial, and ArduinoJson.

2. Upload Firmware: Upload the SWMT_arduino_code.ino sketch to the Arduino board.

3. Core Logic: The Arduino will now continuously execute the following:
    - Read all three sensor values (Temp, Level, Turbidity).
    - Automated Control: Manages Pump 1 based on the water level (waterQty <= 5 starts the pump).
    - Remote Control: Listens for remote JSON commands ({"state":1} or {"state":0}) received from the ESP32 to control Pump 2.
    - Data Formatting: Concatenates all readings and the current pump state into a comma-separated string (e.g., temp,waterQty,isPumpOn,ntu;) and transmits it via SoftwareSerial to the ESP32

### Phase 3: ESP32 Programming (Gateway & MQTT)

The ESP32 is the network gateway, bridging the sensor data to the cloud.

1. IDE Setup and Libraries: Install the ESP32 board package and the required libraries: WiFi, PubSubClient, and ArduinoJson.

2. Configure Credentials: Before uploading, open the SWMT_Esp32_Code.ino sketch and update the Wi-Fi SSID and Password.

3. Upload Firmware: Upload the modified sketch to the ESP32 board.

4. Core Logic: The ESP32 constantly performs the following:
    - Connectivity: Connects to the configured Wi-Fi network and maintains a connection to the MQTT broker (test.mosquitto.org or your production broker).
    - Publishing: Reads the sensor string from the Arduino via Serial2 , parses it , creates a JSON payload ({"waterLevel":..., "temp":..., ...}) , and publishes it to the topic waterwatch/data.

- Subscription: Subscribes to the control topic waterwatch/pumpstate. When a command is received, it forwards the JSON command (e.g., {"state":1}) back to the Arduino via Serial2

### Phase 4: Cloud/Middleware Setup (ThingSpeak)

This phase establishes the backend for data visualization and control.

1. ThingSpeak Setup: Create an account on ThingSpeak (or your chosen MQTT-compatible platform).

2. Create Channel: Create a new Channel on ThingSpeak with fields corresponding to the data published by the ESP32 (e.g., Temperature, Water Level, Turbidity).

3. Configure MQTT: Set up the ThingSpeak MQTT integration. The ESP32 must publish data to the specific ThingSpeak MQTT API endpoint and format (typically using a Write API Key).

4. Verify Data Stream: Check your ThingSpeak channel to ensure data is correctly flowing in from the ESP32 under the waterwatch/data topic.

### Phase 5: Application/Dashboard Building

The final step is creating the user interface (Application Layer) for interaction.

1. Dashboard Development: Develop the web or mobile application (e.g., using React or a similar framework mentioned in the architecture).

2. Data Visualization: Link the application to the ThingSpeak API to retrieve and display the data for all monitored parameters (Liters Used, Temperature, Turbidity, Water Level visualization).

3. Control Interface: Build the Pump Control interface (e.g., an ON/OFF button or toggle).

4. Implement Actuation Logic: Configure the control interface to publish a JSON command ({"state":1} for ON, {"state":0} for OFF) to the MQTT topic waterwatch/pumpstate. This command is then received by the ESP32 and forwarded to the Arduino to actuate Pump 2

# FUTURE SCOPE

The incorporation of **predictive analytics** into the SWMT system provided users with forward-looking insights that enhanced water management. The results in this area include:

**Accurate Predictions**: The machine learning algorithms developed for the SWMT system accurately predicted water usage patterns and potential shortages based on historical data and current conditions. These predictions allowed users to plan ahead, ensuring that water storage was managed effectively even during peak usage times.

**Proactive Recommendations**: The system's ability to provide proactive recommendations based on future weather conditions and usage trends enabled users to optimize their water usage in anticipation of changing circumstances. For example, users were advised to fill their tanks before expected high-demand periods or during cooler temperatures to minimize evaporation losses.

---

# CONCLUSION

The Smart Water Management Technology (SWMT) project successfully addressed critical inefficiencies within traditional water management systems. By integrating IoT, cloud computing, and AI, the system successfully delivered an integrated, automated solution to modern water management challenges.

The system achieved its primary objectives, including the accurate, real-time monitoring of water level and quality and the implementation of automated controls that prevented tank overflows. The integration of predictive analytics and an intuitive user interface led to a measurable reduction in water wastage and positive behavioral change among users.

By demonstrating the effectiveness of combining IoT, cloud computing, and AI, the SWMT system offers a scalable model for sustainable water resource management. Future work will focus on expanding the predictive scope and integrating smart leak detection.