ABSTRACT

Chapter 1:

The proposed Water Management System is designed to address inefficiencies in water usage and distribution by leveraging IoT-enabled devices and automation technologies. Sensors, actuators, and a centralized control system form the backbone of this innovative solution. The system monitors water levels, consumption, and quality in real-time while automating processes like irrigation, distribution, and waste management. Future expansions include advanced data analytics and integration with smart city infrastructures for a sustainable water ecosystem.

INTRODUCTION

Chapter 2:

Efficient water management is critical in urban and rural areas to address challenges like scarcity, overuse, and contamination. The Water Management System aims to optimize water resources by incorporating IoT sensors to measure parameters such as flow rates, pressure, and quality. A web-based application allows for remote monitoring and management, empowering users to make informed decisions and reduce wastage.

2.1. Objective

The primary objective is to create a system that ensures optimal water distribution, minimizes wastage, and provides actionable insights for better resource management.

2.2 Problem Statement

Traditional water management methods are often plagued by inefficiency and lack of monitoring. This results in wastage, uneven distribution, and poor quality control. By automating monitoring and distribution, the proposed system tackles these challenges effectively.

APPLICATION

Chapter 3:

The Water Management System has a wide range of applications:

- **Residential Areas**:

  The Water Management System empowers households to reduce water waste through intelligent leak detection and automated alerts. By analysing real-time water consumption patterns, the system can also provide personalized conservation recommendations. Automated storage tank management ensures continuous water availability while preventing overflow or wastage.

- **Agricultural Fields**:

  Advanced irrigation systems monitor real-time soil moisture, weather data, and crop needs to deliver water precisely where and when required. These systems optimize water use, improve crop yield, and reduce dependency on manual irrigation. Integration with satellite imagery and remote sensors enables large-scale farming operations to maintain sustainability and boost productivity.

- **Industrial Applications**:

  Industries benefit from a centralized water management approach that ensures compliance with environmental standards by monitoring water quality and usage. Smart systems optimize cooling processes, reduce effluent discharge, and enable recycling within the plant, contributing to cost savings and environmental stewardship.

- **Public Facilities**:

  Airports, hospitals, and schools can use the system to optimize water usage for facilities like restrooms, landscaping, and sanitation. Automated scheduling and maintenance alerts help reduce operating costs and promote eco-friendly practices.

COMPONENTS

Chapter 4:

The Smart Parking System comprises the following major components :

- **Raspberry Pi** :
  The Raspberry Pi serves as the brain of the water management system, integrating data from various sensors and peripherals to ensure efficient and automated operation. It processes input from water flow sensors, quality sensors, and level sensors, communicates with the Flask server to update system statuses, and controls the operation of pumps and valves. Its compact size, energy efficiency, and ability to interface with multiple devices make it an ideal choice for IoT-based water management applications.

- **Ultrasonic Sensor (HC-SR04)** :
  The ultrasonic sensor measures the distance between the sensor and a water surface to determine the water level in tanks, reservoirs, or other storage systems. It operates by emitting sound waves and calculating the time taken for the echo to return, ensuring precise measurements without direct contact with water.
    - **Key Features:**
      - High Accuracy**:** Detects water levels within a range of 10 cm to 90 cm, suitable for a variety of storage systems.
      - Non-Contact Detection: Prevents wear and tear, ensuring durability even in harsh environmental conditions.

- **Functionality in the System:**
  - Measures the current water level in tanks or reservoirs with high precision.
  - Triggers pumps to refill water when levels fall below a certain threshold.

- **Relay Module** : The relay module acts as a switch to control high-current devices like water pumps or motorized valves in the water management system.
  - **Key Features** :
    - Operates at Low Voltage**:** Accepts signals from the Raspberry Pi to control high-power devices.
    - High-Current Capability**:** Effectively manages devices such as water pumps and motorized valves.
  - **Functionality in the System** :
    - Controls motorized valves to regulate water flow in irrigation or distribution networks.
    - Enhances efficiency by ensuring pumps and valves operate only when necessary.

- **Stepper Motor**:
  The stepper motor plays a vital role in the water management system by precisely controlling the position of valves, gates, or other mechanical components. Its ability to perform incremental movements ensures accurate regulation of water flow and distribution.
  - **Key Features** :
    - **High Precision:** Enables precise control over valve or gate positioning, allowing for fine adjustments in water flow.
    - **Reliable Performance:** Capable of continuous operation without losing position, ensuring consistent functionality.

- **Functionality in the System** :
  - Opens or closes tank inlet/outlet valves based on water level thresholds detected by ultrasonic or float sensors.
  - Maintains optimal storage levels by activating or deactivating water inputs at precise intervals.

- **Web App** : The web app provides a user-friendly interface for managing and monitoring water resources.
  - **Key Features** :
    - **Real-Time Status Monitoring:** Displays real-time data on water levels, flow rates, and consumption across various parts of the system.
    - **Regular Updates via APIs:** Ensures continuous synchronization with the water management system, delivering up-to-date information on water usage, system health, and operational status.

- **Flask Framework** :

  Flask serves as the backend framework for the water management system, handling API requests and processing real-time data from sensors.
  - **Key Features** :
    - **Lightweight and Easy to Set Up:** Simple to configure and deploy, making it ideal for small to medium-scale water management applications.
    - **Supports RESTful APIs:** Ensures seamless communication between the frontend (web app), sensors, and controllers, enabling smooth data flow and system control.
  - **Functionality in the System**:
    - Flask processes commands from the web app, such as controlling pumps, valves, and adjusting water flow, ensuring actions are executed as per user instructions.
    - Flask ensures that system operations (like irrigation, tank refilling, or water flow management) are automatically adjusted based on sensor inputs, maintaining efficient water usage.

- **HTML, CSS, JavaScript** :

  These technologies are used to build the web app interface for managing and monitoring the water management system.

  HTML: Structures the content and layout of the web app, organizing data such as water levels, flow rates, and system statuses.

  CSS: Styles the interface to ensure a clean, user-friendly, and professional appearance, making the system easy to navigate and understand.

  JavaScript: Enables dynamic updates, such as real-time water level changes, pump statuses, and other system parameters.

  - **Functionality in the System** :
    - Offers a user-friendly and visually appealing platform for monitoring and controlling water management tasks, including valve control, water storage, and irrigation.
    - Enables real-time updates on water levels, flow rates, and usage, allowing users to make adjustments and receive notifications as needed for efficient system management.

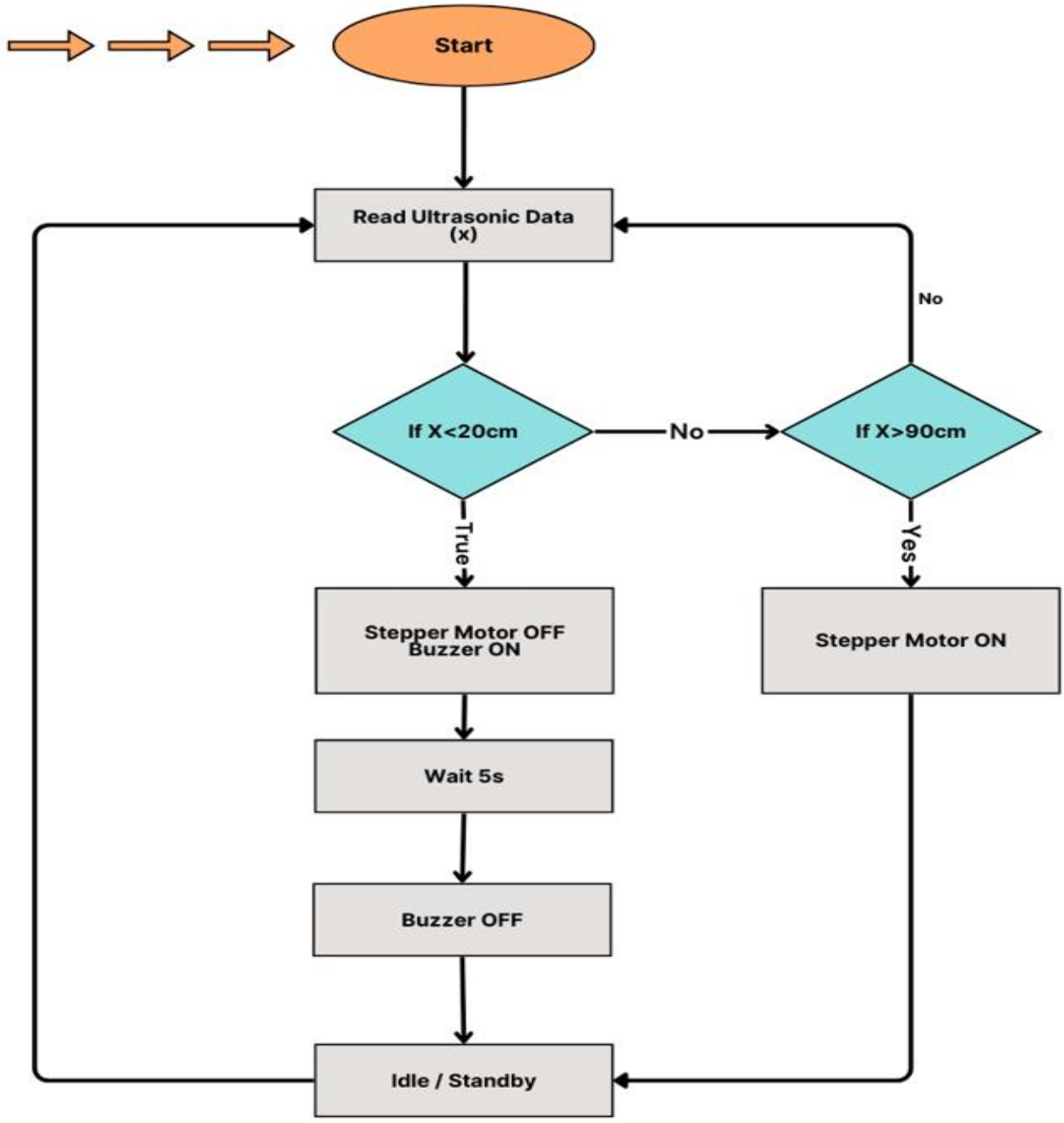
- **Power Supply** :

  The power supply ensures continuous operation of all components, including sensors, pumps, and controllers, within the water management system.

  - **Key Features:**
    - **Stable Output Voltage:** Protects sensitive components from voltage fluctuations.
    - **Compatibility:** Works seamlessly with the Raspberry Pi and other peripheral devices.
  - **Functionality in the System:**
    - **Continuous Power Supply:** Ensures reliable, uninterrupted operation of the system.
    - **Protection from Power Issues:** Safeguards the system against power fluctuations or outages, maintaining consistent performance.

FLOWCHART

Chapter 5:



CONCLUSION

Chapter 6:

This project successfully developed a prototype Smart Water Management System that automates water distribution and monitoring, offering an intuitive web app for real-time updates and control. The system has demonstrated its potential to improve efficiency, conserve water resources, and enhance user experience in both residential and industrial water management applications. By leveraging automation and sensor data, the system ensures optimized water usage, contributing to sustainable resource management.

FUTURE WORK

Chapter 7:

• **Scalability:** To accommodate the growing demand for water resources and system management, the water management system should be scalable. By incorporating additional sensors, expanding data storage, or utilizing cloud-based platforms, the system can handle larger areas, more users, or more complex water distribution networks. This would ensure optimal water distribution and management even in large-scale applications, such as in smart cities, industrial zones, or agricultural fields where water conservation and distribution management are crucial.

• **Mobile App Development:** A dedicated mobile app will offer greater convenience to users by providing real-time monitoring of water usage, allowing remote control of irrigation systems, water storage tanks, and other critical water management components. The app could send push notifications to users about water levels, leaks, irrigation schedules, or water treatment alerts. This mobile solution would increase engagement and improve management efficiency for homeowners, businesses, and farmers by allowing them to monitor and control their water resources from anywhere.

• **Advanced Security Features:** Enhancing the security of the water management system is essential, especially when it involves critical infrastructure like valves, pumps, and water meters. Integrating technologies like RFID, QR codes, or biometric authentication will prevent unauthorized access to these components. This is particularly useful in ensuring that only authorized personnel can manage or modify the water supply settings. Secure access controls will be crucial in commercial, industrial, and urban water management applications, where unauthorized changes or tampering could lead to inefficiencies or system failures.

APPENDIX

Chapter 8:

8.1 GPIO Pin Configuration

- **TRIG (19):** Sends a trigger pulse to the water level sensor for measuring water levels.
- **ECHO (26):** Receives the reflected signal from the sensor to determine water levels.
- **RELAY_PIN (36):** Controls the relay module to activate or deactivate the water pump.
- **ROW_PINS (22, 18, 2, 3):** Represents the rows in the keypad used for manual pump control.
- **COL_PINS (8, 10, 9, 11):** Represents the columns in the keypad used for manual pump control.

8.2 APIs

- **POST /start_pump:**
  Initiates the water pump operation for a specified tank based on the tank ID provided in the request.
  - **Request Format: Requires the tank ID.**
  - **Response:**
    - **Success:** Confirms that the pump has been started.
    - **Error:** s
- **GET/get_water_levels:**
  Fetches the current water levels of all connected tanks.
  - **Response:** Provides the water level status for each tank, such as "full," "half," "low," or "empty."

  Json :

```
{
  "tank1": "full",
  "tank2": "half",
  "tank3": "low",
  "tank4": "empty"
}
```

PSEUDOCODE

Chapter 9:

Sensor Loop (sql)

Initialize:

- Set up GPIO pins and configure the ultrasonic water level sensor.

Start Infinite Loop:

- Trigger the ultrasonic sensor.
- Measure the water level using the echo signal.
- **If water level < threshold**:
    - Detect low water level in the tank.
    - Prompt the user via keypad to start the water pump.
    - **If the user presses 'yes'**:
        - Send a request to the server to activate the pump.
        - Confirm that the pump has started.
    - **If the user presses 'no'**:
        - Continue monitoring the water level.
- **Else**:
    - Mark the water level as sufficient and continue monitoring.

Flask Server (sql)

- **Define Route for POST /start_pump**:
  - Parse the input tank_id.
  - Check if the tank is valid and if the pump is currently idle.
  - **If the pump is idle and the tank is valid**:
    - Update the status of the tank to "filling."
    - Activate the pump.
    - Return a success message indicating the pump has started.
  - **Else**:
    - Return an error message indicating the pump is already running or the tank ID is invalid.
- **Define Route for GET /get_water_levels**:
  - Retrieve the current water levels of all connected tanks.
  - Return a JSON response showing the status of each tank (e.g., "full," "half," "low," "empty").