DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY

SHARDA UNIVERSITY, GREATER NOIDA

TITLE

SMART TANK MANAGEMENT SYSTEM

A project submitted

in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering

by

Rohit Kumar (2019006520)

Shubham. (2019004953)

Supervised by:

Ms. Saptadeepa Kalita (SET Assistant Professor)

MAY, 2023

CERTIFICATE

This is to certify that the report entitled **“Smart Tank Management System”** submitted by **“Rohit Kumar (2019006520) and Shubham (2019004953)”** to Sharda University, towards the fulfillment of requirements of the degree of **“Bachelor of Technology”** is record of bonafide final year Project work carried out by them in the **“**Department of Computer Science & Engineering, Sharda School of Engineering and Technology, Sharda University**”**.

The results/findings contained in this Project have not been submitted in part or full to any other University/Institute forward of any other Degree/Diploma.

Signature of the Guide

Name: Ms. Saptadeepa Kalita

Designation:(SET Assistant Professor

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh

Place: Sharda University

Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honoured to have so many wonderful people lead us through in completion of this project.

First and foremost, we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

Our grateful thanks to Ms. Saptadeepa Kalita for her guidance in our project work. Ms. Saptadeepa Kalita, who in spite of being extraordinarily busy with academics, took timeout to hear, guide and keep us on the correct path. We do not know where we would have been without her help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students:

Rohit Kumar (2019006520)

Shubham (2019004953)

ABSTRACT

The water tank is a vital part of every household. But unfortunately, it can also be a major source of wastage. If we talk about Traditional water tanks which is not able to monitor the water level in water tank and not able to control the water pump, thus wasting of water due to overflowing of water is very serious problem. Various technology may have had issues of one kind or other. This project's main goal is to solve these issues and offer an effective, affordable solution. In this research, an IoT (Internet of Things) system that aids users in solving such issues is introduced. Our system is applied to monitoring the water levels and managing our system using IoT and Web applications. The accessible Iot technology we implement is Arduino. The water tank's level of water is broken down into maximum, minimal, and nominal levels, each of which is indicated by percentage of fulfilment of water. We Install an HC-SR04 ultrasonic sensor on the tank's top to monitor the water level, measure the distance between the point and the sensor, and transfer the information to the Web application via ESP32. The Web application offers a user interface with buttons for manual operation, a motor status LED, and a tanks design diagram indicating percentage of water level.

CONTENTS

LISTOFFIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem statement

The overflowing of water tanks is a common issue that causes water wastage and can also lead to water pollution. The conventional method of manually monitoring the water level in tanks is inefficient and can lead to human error. This issue can be tackled using technology that can automate the process of monitoring and controlling the water level in tanks. The capacity of the existing water supply infrastructure to meet forecast water supply is related to the water level supply problems identified within the region. Nowadays, owing to our hectic schedules, we frequently overlook little details such as turning off the water pump, and as a result, we are experiencing issues such as water and energy shortages. The objective of this research is to develop a water level controller system that can automatically control the water level in tanks and prevent wastage of water and electricity. Overall, the development of an effective and affordable system for monitoring and controlling water levels in tanks has the potential to make a significant impact in reducing water waste and preventing pollution, particularly in areas where access to clean water is limited.

1.2 Project Overview

Water is the most important thing found on earth which is helpful for living beings to survive. Without water, life is impossible on earth. It is the most precious thing for living beings. Water conservation is very essential in order to preserve the groundwater. Due to the population explosion and their growing aspirations for urbanisation, industry, and other forms of development, safe drinking water is becoming more polluted and harmful. The oceans hold 97% of the world's water, which covers 71% of the planet. The ocean water is too salty and cannot be used for domestic purposes like drinking, cooking, washing, and so on. So, only groundwater (3% of a total volume of water) is accessible. Which is used by humans to fulfill their daily needs. In recent surveys, it is found that freshwater resources are rapidly diminishing, Creating panic among various countries. The primary causes of this crises include (1). Uneven water management deployment (2). consumption of water increased drastically in urban areas (3). Lack of awareness among humans in order to consume water resources. In rural areas, water is largely used from ponds, lakes, and tube wells by the people to fulfill their agricultural uses, personal hygiene, washing, drinking, watering plants, etc. Whereas in the urban areas, water distribution by the government bodies like municipal corporation large overhead or underground water tanks are used to store the water, and various parts of major cities receive water supply from such tanks.

There is an abundance of water, particularly in small towns and cities. However, although the population of the planet is expanding, there is a shortage of freshwater resources. The need for water is rising today as a result of population expansion in both urban and rural areas. Naturally, the safe drinking water distribution system must be maintained in order to consistently meet people's needs for safe drinking water as that the consumption of water rises. Fresh water is desperately needed, particularly in crowded cities.

Water tanks are one of the key pieces of equipment that almost every local, state, and private institution has today as a storage container for clean water for daily use. It is well known that the availability of clean water is a critical part of safety, comfort and energy efficiency for customer satisfaction and productivity. Water tanks on the roof/tower require pumps as a means of filling the water, and its distribution uses gravity to direct the water to multiple points of use, so the water tanks on top require no power or other power. More likely to be chosen to avoid things... blackouts.

1.3 Expected outcome

A water shortage is a serious problem that needs immediate attention. Cities, towns and countries face a severe water shortage in the future if they don't take precautionary measures. A smart water management system helps manage water scarcity. It uses sensors, data collection, and management algorithms to conserve water. The system can be made using the ESP32 and Arduino IDE platforms. IOT devices use less water because of their efficiency and water-saving features. The project enables user to manage to effectively monitor water level using ultrasonic sensor such as temperature sensors, relative humidity sensors. At regular intervals (5 seconds), the sensor collects information about the water tank and water, which is recorded and stored online using ESP32 server. The recorded data is transferred to the web server database via wireless transmission. A user can remotely monitor and control the system using a web application that presents her web interface to the user.

1.4 Hardware & Software Specification

1.4.1 Hardware

- **ESP32-** Modern microcontrollers should take the place of Arduino now. Today, we begin with the well-liked ESP32. Despite having many excellent qualities, the Arduino lacks in several areas. The first is speed. Most Arduino AVR series boards operates at a 16MHz frequency. While this is undoubtedly quick enough to construct numerous applications, it becomes a barrier toward others.

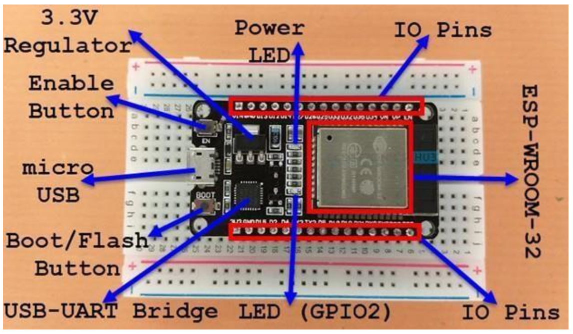

Fig:1.1 ESP32

- **Ultrasonic Sensor:** Ultrasonic sensors (or transducers) work on the same principle as radar systems. Ultrasonic sensors can convert electrical energy into sound waves and vice versa. Acoustic signals are ultrasonic waves propagating at frequencies above 18 kHz. The well-known ultrasonic sensor HC SR04 produces ultrasonic waves with a frequency of 40 kHz.

Figure: 1.2 Ultrasonic Sensor

- **Resistor:**

  Resistors are electronic components that regulate or limit the flow of electrical current in a circuit. They are often used to provide a specific voltage for an active device, like a transistor. By impeding the flow of current, resistors can help to prevent damage to sensitive electronic components. They are available in different types, each with a different resistance value, and can be connected in series or parallel to achieve a desired resistance. Resistors are an essential component in the design and operation of electronic circuits.

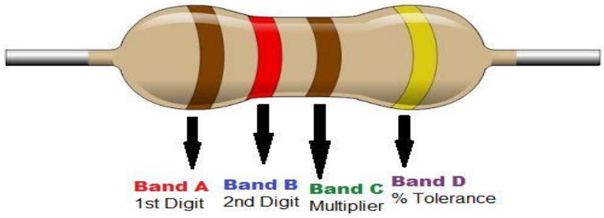

Figure: 1.3 Resistor

- **IC 7805-** IC 7805 devices are used as power supplies in circuit. These ic 7805 devices have 4 pins that can be connected to external power sources or to heat sinks. Additionally, these IC 7805 devices are used in electronic project schematics and circuits. They are also used in industrial controls, medical equipment and food storage. A lot of ic 7805 devices are manufactured and sold by various companies; this is why there are many applications for these devices.

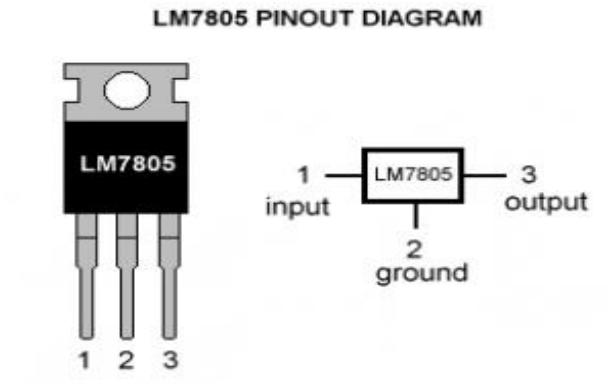

Figure: 1.4 IC7805

- **Capacitor-** It is a set of collinear conductive material that serves as an electrical energy storage device. A simple example of this type of storage and handling is plate capacitor. If there is a positive charge of total charge +q is carried out on one of the conductors and the same negative charge -Q is deposited on the electrically conducting, they are referred to the charge Q of a capacitor. (Also see Absorption Theory for Electricity.) Capacitor use is common in embedded services. These are used in electrical circuits to safeguard data stored in large system memory from loss due to a fleeting power interruption.

Figure: 1.5 Capacitor

- **Humidity Sensor-** Humidity is the measure of the amount of water vapor present in the air. Humidity is calculated as Relative humidity and Absolute humidity. For industrial and medical environments relative humidity becomes an important factor. A rise in the values of humidity, beyond threshold levels, can lead to malfunctioning of control systems, errors in weather prediction systems.

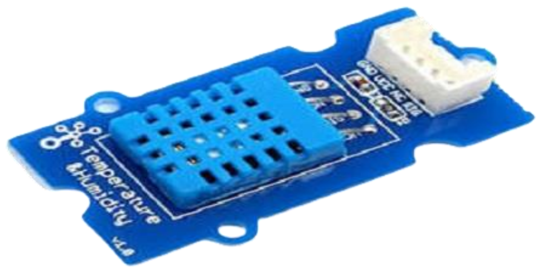

Figure: 1.6 Humidity Sensor

1.4.2 Software

The software used in this project includes Arduino IDE and HTML/CSS programming languages. Arduino IDE is an open-source integrated development environment (IDE) used to write, compile and upload code to Arduino boards. It is based on the C++ programming language and allows easy programming of microcontrollers, such as the ESP32 used in this project. C++ is a high-level programming language that is commonly used in embedded systems and software development. In this project, C++ is used to set conditions for the water level controller system. This includes setting the range of the ultrasonic sensor, establishing the critical water level threshold, and programming the ESP32 microcontroller to turn on/off the water pump based on the water level in the tank. The C++ code is compiled and uploaded onto the ESP32 microcontroller using the Arduino IDE software.

In addition to the Arduino IDE, we also used HTML/CSS programming languages to create a web application for the water level controller system. HTML (Hypertext Markup Language) is a markup language used to create the structure and content of web pages, while CSS (Cascading Style Sheets) is used to style and format the appearance of these pages. HTML (Hypertext Markup Language) is used to structure the content and layout of the web page. It defines the different elements on the page, such as buttons and text fields, and their placement. CSS (Cascading Style Sheets) is used to define the visual styling of the web page, such as colours, fonts, and backgrounds. Together, HTML and CSS create an interactive user interface that allows the user to control and monitor the water level controller system remotely. The combination of these programming languages allowed us to create a user-friendly interface for controlling the water level controller system through a web page. The Arduino IDE was used to program the ESP32 microcontroller to communicate with the sensors and control the pump, while the HTML/CSS web application allowed the user to view and interact with the system's data and settings.

The web application is hosted on the ESP32 microcontroller using the ESPAsyncWebServer library. This library allows the ESP32 to act as a web server, which means that it can serve web pages to any device connected to its network. The user can access the web application by connecting to the ESP32's Wi-Fi network and entering the IP address of the ESP32 into a web browser. Once connected, the user can control the water level controller system and monitor its status in real-time.

Software and languages used are following:

- Html
- CSS
- C++
- Arduino IDE

Chapter 2

LITERATURE SURVEY

2.1 Existing Work

They develop a system in which they use many IoT devices for Smart water pump Switches. They use IC555, Relay board, Crystal Oscillator, Switch, and wires. The working of ic555 and the crystal oscillator is sensing the water level and transmit the signal to the electric circuit board *(Relay Board)* that is connected through the wire to the water pump. After receiving the signal, they act out their specific work like on/off the water pump [1].

Water level monitoring system which is an IoT device. They design this system because they can control the water wastage which is occurred by the overflowing of the water tank in an urban area. This technique aids urban residents in determining the water level and prevents drinking water waste in cities. [2].

In this paper, they look into a water monitoring system's architecture. They then created a system that can automatically monitor the water level and operate the water motor. Their technology first detects the water level before sending the signal to a board that is connected to the relay board. With the use of flip-flop sequential logic, this device automates processes. When the water level reaches the water tank's lowest level, this device automatically turns on, and when it reaches its highest point, it automatically turns off. [3].

They developed a water monitoring system using gsm technology. They design a system that uses gsm communication technology. They monitor the water level and send the signal through SMS to the person. Their prototype is tested and implemented over many areas. This is a straightforward GSM network-based water level monitoring device. [4].

The water level is sensed by this system's ultrasonic sensor for overhead and underground tanks. Additionally, this will include a pump switching and control system that displays and tracks the tank's water level. Power supply, microcontroller, sensor, display, and Arduino Uno make up this system. They use a 20 kHz ultrasonic sensor to determine the water level, and a crystal LCD display to show it. [5].

They deliver a study on Internet of Things (IoT) devices that assist users in monitoring and managing water use. The device is simple to put anywhere in the residential area, and we can see the water level there. due to the fact that the data will be cloud-stored. The water motor is automatically controlled by the motor in real-time in accordance with the water level. When they fill the tank to the top, the motor will automatically turn on and shut off when they reach the bottom. [6].

The ZR16S08 microchip is described in this paper as an Internet of Things (IoT) alternative for a smart water system. Knowing that leakage is one of its defining characteristics, the systems carefully check the water flow in the distributing channel's pipelines to assure the quality of the water supply. [7]

An important key characteristic of the IoT water management system is automation. Everyone's life is made simpler and easier as a result. This is something we'd like to provide in the context of water conservation. Our goal is to educate people about valuation of conserving water and the operation of water-management systems become possible by the Internet of Things (IoT). is to [8]

A platform-independent, service-oriented architecture called OPC UA (Object Linking and Embedding for Process Control Unified Architecture) is used to monitor procedures of manufacturing and distribution. On the basis of this concept, we provide a paradigm for smart water distribution that fuses Internet of Things technology with technologies for coordinating business processes and making decisions. They provide detailed descriptions of the architecture for subsystem interactions and the physical scenarios against which the implementation is tested to manage and interoperate with specific vendor equipment in specific contexts of water management processes. [9]

Groundwater had already long played a significant role in daily living. Water management and protection are necessary for human economic stability given the situation of the environment globally. Consumer-based humanitarian projects that may be quickly built using Internet of Things (IoT) technology are greatly needed nowadays. This paper implies a real-time, Internet of Things-based system for water monitoring. Our prototype is based on the notion that, particularly in blunder areas, level of water is a crucial element when coping with floods. A water level sensor to monitor a desired parameter, and when the parameter is reached, a signal is immediately forwarded to social media sites like Twitter. Data storage was set up on a cloud server. Readings for the water level are shown on the screen of the transmitter. [10]

2.2 Proposed System

The proposed system is designed to address the problem of water tank overflow and the resulting water pollution. It is a water level controller system that uses an ultrasonic sensor to monitor the water level in the tank and control the water pump accordingly. The system consists of an ESP32 microcontroller, an ultrasonic sensor, a water pump, and a buzzer. The ESP32 microcontroller is used to interface with the ultrasonic sensor and control the water pump. The ultrasonic sensor is used to measure the water level in the tank, and the microcontroller uses this information to turn the water pump on or off as needed. The buzzer is used to alert the user when the water level reaches a critical point.

In addition to the basic functionality of controlling the water level in the tank, the system also includes a web-based user interface that allows the user to monitor the system and control the water pump remotely. The web-based user interface is built using HTML, CSS, and C++, and is hosted on the ESP32 microcontroller.

The proposed system is designed to be easy to install and use, with minimal setup required. It is also designed to be low cost and energy-efficient, making it accessible to a wide range of users. Overall, the system is an effective solution to the problem of water tank overflow and can help to prevent water pollution while saving water and energy.

Chapter 3

SYSTEM DESIGN & ANALYSIS

This study presents a novel automated water tank system and pump control system that aims to regulate water levels and prevent overflowing of water from the overhead tank. The system utilizes an ultrasonic sensor, humidity and temperature primary sensors, with ESP32 serving as the microcontroller unit. The ESP32 is programmed to monitor the water level, temperature and humidity in the tank which is on overhead of buildings and automatically turn on the pump if it falls below a predetermined threshold. An additional buzzer notifies the users when the level of water is low. If the level of water reaches to a critical level, designed system activates the pump, which begins pumping the water until the water tank is full. The water pump then turns off automatically to prevent overfilling. Overall, this system provides an efficient and effective solution for managing water levels in an overhead tank, reducing the risk of overflow and ensuring a steady supply of water.

The proposed system includes a web application that displays important information such as the pump status, temperature of water and humidity of surface of water tank in the overhead t. The web application also automatically displays a message indicating that the level of water in the overhead tank is below the defined level, and many more information like status of the water pump is off/on. when the water tank will be empty then water pump is automatically turned off. This method helps to prevent water wastage due to overflowing of the tank and control electricity consumption by avoiding the running of the pump unnecessarily.

The proposed system is designed to conserve electricity and prevent water wastage. It is set up to automatically control the water levels in the overhead tank, ensuring that there is no overflow and no running of the pump when not needed. The system uses an ESP32 board, which is connected to the sensors and work properly as it is programmed. The ESP32 is directly connected to the circuit and controls the electricity flow in the water pump circuit, ensuring that it only turns on when the level of water reached up to a minimum level. There are several sensors used in the system, including an ultrasonic sensor, humidity sensor, and temperature sensor. The ultrasonic sensor measures distance by emitting and receiving ultrasonic waves, and it is used to measuring the level in the tank. The humidity sensor measures the humidity of the tank surface and can detect the causes of water pollution. The temperature sensors used to measure the temperature of the water, which is important for monitoring water quality. The

ESP32 operates at 5V and controls the current flow to the water motor, which is directly connected through the circuit of the designed system. This system is very helpful in the conservation of electricity and prevents the wasting of water by controlling the levels of water.

Overall, Proposed system is designed to be efficient and effective in monitoring the water levels and preventing wastage by overflowing of water. By using sensors to detect the water levels and other important information, the system can save electricity and prevent water overflow, ultimately leading to more sustainable and responsible water usage.

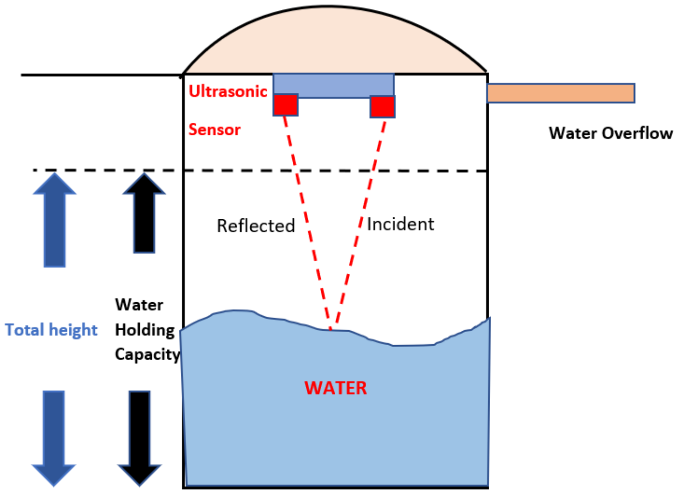

Figure: 3.1 Framework of Proposed system

The project involves connecting the echo pin to the 11th pin and the trigger pin to the 12th pin of the ESP32 board. Additionally, the LCD is connected to the ESP32 board using specific pins such as AS=B5, F=B4, C4=B3, C5=B2, C6=B1, C7=B0. The ESP32 board is designed to detect ultrasound within the range of 7cm to 40cm, which is suitable for detection of the level of water in the water tank with a depth of 20cm. When the water reached at minimum state in tank, the pump buzzer will start buzzing at 20cm to 16cm of the water detection in the tank. At 7cm from the top, the ESP32 board will turn off the motor as it considers that the overhead tank to be full.

3.1 Working

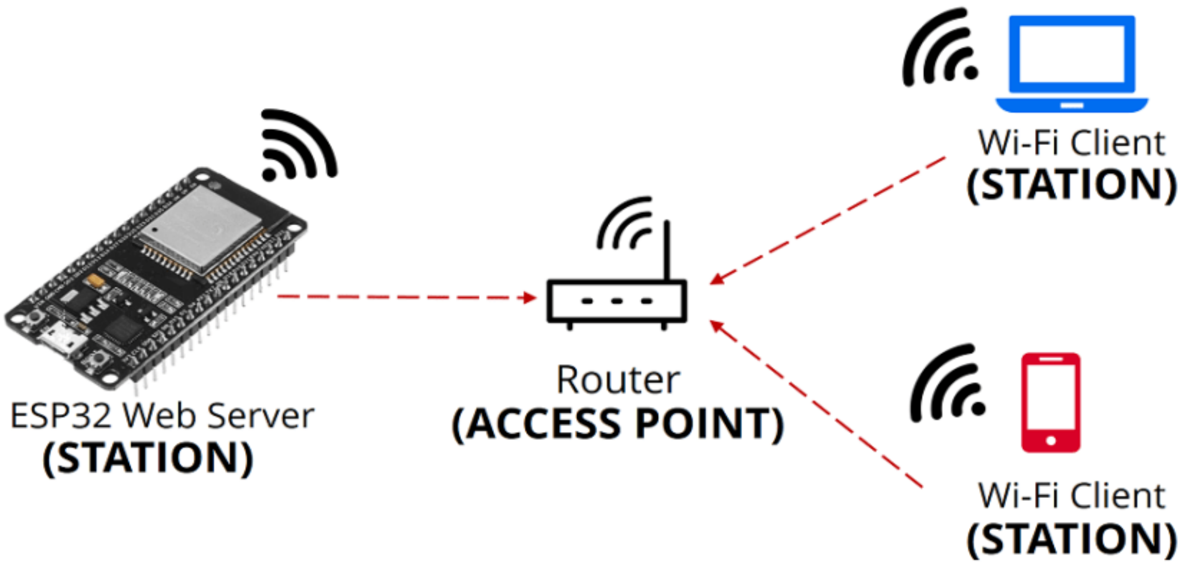

Figure: 3.2 Working Principle

The ESP32 can function as a Wi-Fi access point, station, or both. When working with the ESP32 in projects, it is common to connect it to a Wi-Fi router so that it can be accessed over a local network. To achieve this, the ESP32 is set as a station and connected to the router, which acts as an access point. By doing so, we can control the ESP32 through a local area network.

STEP 1) SCAN WIFI NETWORK (FIND ROUTER OR HOTSPOT)

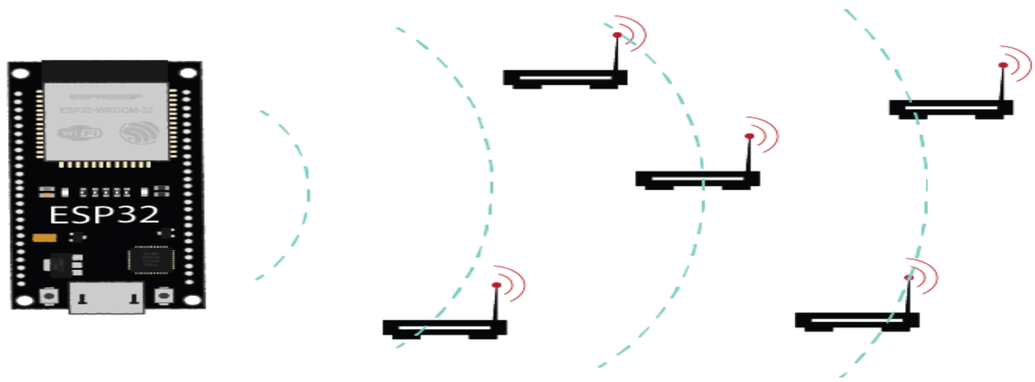

Figure: 3.3 Scanning Wi-Fi router

It will scan all the nearby networks and connect to the matching ssid and password. In our case its project and password are project@123

```
// Replace with your network credentials
const char* ssid = "project";
const char* password = "project1234";
```

So, all the devices connected with the hotspot named project can access our ESP32 dashboard.

STEP 2) CONNECT TO WIFI NETWORK

Once you find a network that matches your password and name, the second step is to connect to ESP32. After connecting ESP32 we have triggered a while loop at Arduino IDE platform that repeats until Wi-Fi is connected. Once the Wi-Fi is connected, you will see the local IP address of the connected Wi-Fi and you can access the Dashboard from that IP address.

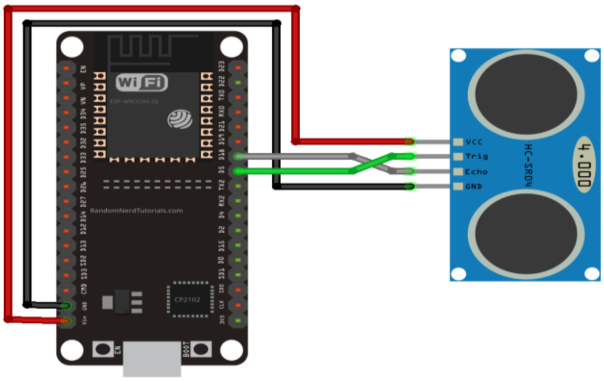

Figure: 3.4 Ultrasonic sensor with ESP32

The ultrasonic sensor (HC - SR04) allow us for 2cm - 400cm non-contact measurement, and distance measuring with accuracy reaches 3mm. This module contains a transmitter, receiver, and control circuitry of ultrasonic. A burst of sound waves is produced when an electrical impulse is generated by applying a high voltage to an ultrasonic transducer vibrating in a specific frequency spectrum. When any obstacle comes in front of the ultrasonic sensor, the sound wave is reflected as an echo and a voltage pulse is generated. Calculate the sound transmission interval between the 5V wave and the received echo. Operating voltage.

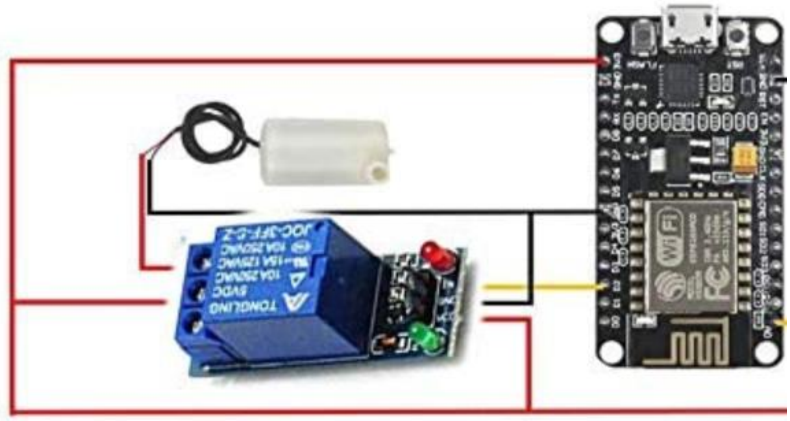

Figure: 3.5 Circuit of Proposed system

we have made the web page which is designed in HTML, CSS, Bootstrap and JavaScript used in designing and JavaScript for client-side interaction such as input the value and interaction with button etc. The project has 2 modes for irrigation auto and manual. In Manual irrigation user have manage their system manually but when user select the automatic irrigation for controlling the water pump the user have not to interact with water pump. Water pump automatically shut down when water tank at the level of overflowing.

3.2 System features

Water management

Water management is a critical aspect of any water tank system, and a smart tank management system can provide users with a range of features to manage their water supply efficiently. The system should be designed to prevent overfilling of the tank, which can lead to water wastage and damage to the tank. In addition, users can monitor the level of water using their smartphone, and the water motor can be operated automatic with this system. When the water goes down, the water motor pump automatically turns on, and when the level is high, the motor is to be shut down as result of this proposed system.

By incorporating these features into the system, users can be assured that their water supply is managed efficiently, with proper water usage and effective conservation measures. The water management system should include sensors that monitor the level of water in the overhead tank and provide an alert when the level of water is low. This information can be used to determine water usage patterns, identify leaks or other issues, and ensure that the water supply is maintained at optimal levels.

Water Monitoring

Water monitoring is a key feature of a smart tank management system, and it can provide users with valuable information about their water usage patterns. The system should include sensors that continuously monitor the level of water in the overhead tank and provide real-time information to the user. This information can be used to determine water usage patterns and identify leaks or other issues. In addition, the system can track water usage over time, allowing users to monitor their water consumption on a weekly basis. By knowing how much water is being consumed, users can take steps to reduce their usage and conserve water. Our app will alert users when their usage is beyond a certain limit, highlighting the importance of water conservation and encouraging them to take action to reduce their water consumption.

Overall, water monitoring is a critical component of a smart tank management system, as it allows users to track their water usage and take steps to reduce their consumption, leading to better water conservation and more efficient water management.

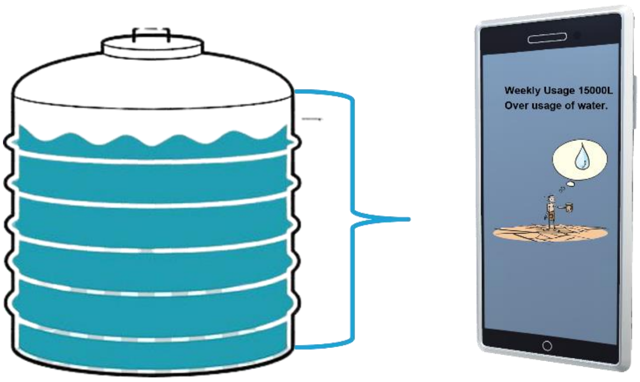

Figure: 3.6 Water Monitoring feature

Remote Controlling

In Remote controlling is a crucial feature of a smart tank management system, allowing users to control the water pump remotely using a web application. With this feature, the user can easily turn the water pump on or off, without the need to be physically present near the tank. In our proposed system, remote controlling is a highlighted feature, and this system allows the admin to control the water motor with ease. User only needs to turn on the water motor using the web application, and the motor will automatically turn off when the level of water touches the highest level of the tank. This feature ensures that the water pump operates efficiently, without the need for constant monitoring by the user.

By incorporating remote controlling into the system, users can manage their water supply more effectively, leading to better water conservation and more efficient water management. The web application is easy to use and provide the user with real-time information about the water level, temperature and surface humidity of the tank, allowing them to make informed decisions about when to turn the water pump on or off.

Humidity Monitoring

Humidity monitoring is a key feature of a smart tank management system, and it can provide users with valuable information about their water quality. The system should include sensors that continuously monitor the level of the water and pH of water. After that It provide the real-time information to the user. This information can be used to determine water quality and identify any contamination or other issues.

3.3 Methodology

The Smart Tank Management System utilizes an ESP32 that is the core hardware. This system features are a powerful Wi-Fi module that can either host a web application or allow for information transfer using Wi-Fi. The ESP32 is operated on the voltage of 3.6V and has onboard mechanisms and storage capacity that allow for coordination with the sensed data. Programming for the ESP32 is done with the Arduino IDE and the C language. Real-time data is displayed on a dashboard, including a graph, gauges, and the volume of water at a specific time. The system has checks for the level of water, which trigger a set of pre-programmed courses of action. The system has two modes of operation, manual and automatic, with an LED to inform the user if the level of water is below or above the marked value. If the level of water is not touching the highest marked level until then, the pump will automatically turn on, and if it is above the threshold, the pump will stop, triggered by the LED.

This system is based on ESP32 IOT module which includes the following steps:

> **Core Hardware Selection:** The primary hardware used in the system is ESP32. It has a powerful Wi-Fi module that allows data transfer through Wi-Fi. It is programmed using Arduino IDE and is equipped with enough onboard storage and mechanisms to coordinate with sensors.
>
> **Programming**: C language is used in Arduino IDE for implementing the code in ESP32. Real-time data is visualized in the dashboard in the term of graph and gauges representing the flow and volume of water present in the tank.

Water Level Monitoring: Two checks are performed in the system - water level inside the tank and water level inside the pump. These checks lead to the execution of pre-programmed courses of action.

Manual/Automatic Control: The system offers two options for controlling the water pump - manual and automatic. In the automatic mode, if the water level in the tank is below the minimum set threshold value, the LED glows to inform the user and automatically turns on the water pump. If the water level in the tank crosses the maximum set threshold value, the LED is triggered to inform the user and stop pumping the water in the tank.

Data Alerting: The system alerts the user through the app when the water usage is beyond the limit or when the contamination level is higher than the fixed threshold.

Overall, the methodology includes the selection of appropriate hardware, programming using Arduino IDE, monitoring water levels, and controlling the water pump. The system also includes an option for manual/automatic control and data alerting to ensure proper water management.

3.4 Evaluation of Water Monitoring techniques

There exist various water monitoring techniques that were implemented since ages. Some of them were the ancient traditional monitoring techniques which later on switched to smart monitoring techniques.

3.4.1 Traditional monitoring

The traditional monitoring water system is a manual process. It involves checking the water tank on a regular basis to ensure that it doesn't overflow. This allows for 24/7 h monitoring of tanks without the need for manual checking. Which is manually may not be feasible in many cases. If any leaks occur on the tank, then they will be reported manually. Which are causes of water wastage.

However, there are various drawbacks to this process. Firstly, it is time-consuming and requires a lot of energy. Secondly, it is not accurate and can lead to the wastage of water as well as electricity. Thirdly, it does not provide alerts when the tank is about to overflow which can result in major problems like flood or leakage in the house. 24/7 Water Monitoring is expensive and difficult for homeowners. There is no water tank that can be monitored 24/7 on its own without needing human interaction to fill it when it reaches a certain level [11].

3.4.2 Offline automated monitoring

Today, there is an emerging trend of water conservation and reforestation. With the increase in population, the growing need for potable water is inevitable. The current method of refilling tanks is not sustainable and inefficient as it requires constant monitoring to refill tanks when they become empty. The system is designed for remote controlling and monitoring of water tanks. The system includes a wireless sensor node, a set of actuators, and a processor. The wireless sensor gathers the data from the sensor and sends it to the processor wirelessly. The actuator sets the tank's valve according to the data received by the processor.

No matter how clean and well-maintained oner water tank is, it is important to regularly monitor the level of residual water for the sake of safety. The offline Monitoring system can automatically measure how much water is left in oner tank (in litters and gallons), and alert one when one need more. This can be used in places where there is no electricity or internet connection available but there is still a need for water storage tanks to be monitored periodically [11].

3.4.3 WSN-based Monitoring

WSN is a new type of wireless sensor network that has been widely used in monitoring water quality in rural areas. Wireless sensor networks are a promising technology for extending the capability of manual monitoring systems. This refers to a network of sensory nodes that are spread out in different locations and operate independently to monitor various physical or environmental conditions. These conditions may include things like temperature, sound levels, pressure, motion, or pollutants in the surrounding area. The nodes work together to gather and transmit data about these conditions for analysis and monitoring. In this system, sensor node reads the sensor data and processes them for monitoring. The captured data sends to the main section through wireless channel (e.g., Lora WAN, Xbee, Wi-Fi, Bluetooth, server). The main channel further processes the data and then send back to the main section. The main section has further functions such as analysis and decision-making. The WSN-based system is a new type of monitoring system that can be used in many different fields, such as agriculture, environment, home security, and health care. This system is based on rural drinking water source monitoring system design can provide timely information for users about water quality in real-time [12].

3.4.4 Smart Monitoring:

It is designed to ensure water conservation by monitoring the quality of water and the amount of water used. This monitoring system consists of sensors, a communication module, a data-logging device, and an application. The sensors monitor the level of dissolved oxygen (DO), pH level, temperature, conductivity (EC), turbidity (NTU), and chlorine residual (ppm) in real-time. The communication module sends data to the cloud for storage and analysis while the data logging device stores all sensor readings in local memory for offline analysis. The application provides users with real-time alerts on potential problems such as leaks or low levels which require attention from maintenance personnel. With the help of a smart water meter, one can monitor oner tanks on oner own schedule- at any time or interval one chooses. One will also not have to worry about detecting leaks in time to stop them because the meters themselves self-regulate the amount of water M. Kumar Jha et al. provide based on consumption patterns detected within the system [13].

3.5 Benefits of IoT Architecture in water management

Below mentioned are some of the benefits of monitoring water level based on IoT architecture.

Introduction of Technologies: In recent years, the Internet of Things (IoT) and related technologies have revolutionized various industries, including water management. IoT architecture involves the use of interconnected sensors, devices, and data analytics tools to monitor and manage water systems in real-time. By leveraging this architecture, water utilities can improve efficiency, reduce costs, and enhance customer service. IoT technology is particularly useful in water management, where it can help utilities to monitor water quality, detect leaks and breaks, and optimize water usage. Sensors can be used to monitor water levels, flow rates, and pressure, allowing for early detection of issues and more efficient use of water. Real-time data analytics can provide insights into patterns and trends in water usage, which can inform decision-making around resource allocation and system optimization. Moreover, IoT technology enables remote access and control of water systems, reducing the need for on-site visits and improving system maintenance and efficiency. This also allows water utilities to provide real-time information to customers on their water usage and billing, improving customer satisfaction and reducing complaints[15].

Real-time monitoring: IoT architecture allows for continuous monitoring of water systems, including water levels, flow rates, and water quality. This enables early detection of any abnormalities or issues in the water system, allowing for timely interventions to prevent water wastage and system failures

Data-driven decision-making: With IoT architecture, data on water systems can be collected and analysed in real-time. This allows for data-driven decision-making, including identifying patterns and trends in water usage, predicting future demands, and optimizing water supply.

Remote access and control: IoT architecture enables remote access and control of water systems, allowing for adjustments to be made to water levels, flow rates, and quality parameters from a central location. This reduces the need for on-site visits and improves system maintenance and efficiency.

Cost savings: IoT architecture can help reduce operational costs by improving water efficiency and reducing water wastage. It can also help with predictive maintenance, which can prevent equipment failures and reduce repair costs.

Improved customer service: With IoT architecture, water utilities can provide better customer service by providing real-time information on water usage, billing, and alerts for water quality issues. This can improve customer satisfaction and reduce complaints.

3.6 Implementation of IoT in water management system

IoT based smart water monitoring system is now implemented in every industrial as well as agricultural sectors. Some of the areas where applications of IoT based water monitoring techniques are used are discussed in the below section.

3.**6.1 Smart Water System**

The water system is a complex, large-scale system. The system includes all the water sources, such as rivers, lakes and streams. All these sources are connected to the water treatment plants that provide clean drinking water to the consumers. As we can see, there are many risks in this process. For example, if someone leaks a pipe and it is not detected in time, then the entire water system will be contaminated with dirty and polluted water. This can lead to outbreaks of diseases that could affect people's health. IoT technology provides an opportunity for detecting these problems before they occur. IoT sensors can detect any changes in the quality of drinking water or even pipe leaks before they have an impact on human health or on the environment A smart water system is a system in which the water quality is monitored by using IoT technology.

3.6.2 Smart Irrigation System

The agricultural industry is already aware of the advantages that IoT technology brings to their sector. Agriculture is one of the sectors that are adopting new technologies to improve productivity and profitability. IoT devices Collects the data that can be used for various purposes such as monitoring and controlling agricultural processes. Weather stations are an example of an IoT device used in agriculture; they collect information from different types of weather sources such as temperature and precipitation. Collected data is then transferred to a database where it's analysed for agricultural forecasting purposes.

Another interesting application of IoT in agriculture is in the field of prediction. Farmers use IoT to collect data from their systems and systems they are connected with via the internet. Data collected can help predict weather conditions before sowing or harvesting crops. Farmers can use this information to schedule their activities regarding crops so they can maximize productivity and profitability while minimizing risk exposure. Some example devices used for agricultural forecasting include solar-powered environmental sensors, digital scales, thermometers, barographs and precipitation collectors. Additionally, farmers can also use mobile terminals to check market conditions, prices and other relevant factors affecting their business decisions. The industry needs new innovative technological solutions for more efficient agriculture operations; this includes implementing smart technology such as the Internet of Things into farms. Many applications like weather forecasting or managing livestock help produce better crops with minimal input from farmers. Implementing smart technology allows farmers to spend more time focusing on producing high quality crops rather than maintaining equipment themselves[14].

3.6.3 Smart Gardening

Gardening is an outdoor activity enjoyed by many people. Many people find gardening therapeutic and enjoy the fresh produce it provides. However, gardening is a time-consuming task; it requires planning, equipment and labour. Using IOT in smart gardening reduces the time spent in tasks and makes them more convenient. Gardening involves a lot of time spent walking around and checking various areas of oner garden. Checking oner garden's progress over time collects data that helps one plan oner garden. An IOT device can be used to collect data about plants, pests and soil conditions. One can then make informed decisions about oner garden's care. Additionally, one can share oner garden's data with others for feedback or to help one with one's problems.

3.7 Required code for System Design

3.7.1 HTML code for Web application

```
client.println("
<!DOCTYPE html>
<html>

<head>
    <title>Project</title>
    <link rel=\"stylesheet\"
href=\"https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.cs
s\"
        integrity=\"sha384-
Smlep5jCw/wG7hdkwQ/Z5nLIefveQRIY9nfy6xoR1uRYBtpZgI6339F5dgvm/e9B\"
crossorigin=\"anonymous\">
    <meta http-equiv=\"refresh\" content=\"1\">
    <style type=\"text/css\">
        .progress-bar-vertical {
            width: 300px;
            min-height: 400px;
            display: flex;
            align-items: flex-end;
            margin-right: 20px;
            float: left;
        }

        .progress-bar-vertical .progress-bar {
            width: 100%;
            height: 0;
            -webkit-transition: height 0.6s ease;
            -o-transition: height 0.6s ease;
            transition: height 0.6s ease;
        }
```

```
    </style>
</head>

<body style=\"width: 100vw;height: 100vh;\">
    <div class=\"jumbotron bg-primary text-light\">
        <div>
            <h1>Smart Tank Management System</h1>
            <hr> dashboard
        </div>
    </div>
    <div class=\"container\">
        <div class=\"card\">
            <div class=\"card-header bg-primary text-light\"> Tank Level
</div>
            <div class=\"card-body\">
                <div class=\"progress progress-bar-vertical\">
                    <div class=\"progress-bar progress-bar-success progress-
bar-striped active\" role=\"progressbar\"
                        aria-valuenow=\"100\" aria-valuemin=\"0\" aria-
valuemax=\"100\" style=\"height:"); if (distance
                        <=5) { digitalWrite(ledPin, LOW);
client.println("100%;\"><span class=\"\">100% Full</span> ");
                        client.println("</div>
                </div>
                <ul>
                    <li>Water Level :");
                        client.println(distance);
                        client.println("CM <span class=\"text-
success\">FULL</span> </li>
                    <li>Pump Status : OFF</li>
                    <li>Tank Height : 21 CM</li>
                </ul>
            </div> ");
            } else if (distance > 5 && distance <= 7) {
client.println("90%;\"><span class=\"\">90% Full</span>");
                digitalWrite(ledPin, LOW);
                client.println("
        </div>
    </div>
    <ul>
        <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-success\">FULL</span></li>
        <li>Pump Status : OFF</li>
        <li>Tank Height : 21 CM</li>
    </ul>
    </div> ");
```

```
    } else if (distance > 7 && distance <= 9) { client.println("80%;\"><span
class=\"\">80% Full</span>");
        digitalWrite(ledPin, LOW);
        client.println("</div>
        </div>
        <ul>
            <li>Water Level :");
                client.println(distance);
                client.println("CM <span class=\"text-success\">FULL</span>
</li>
            <li>Pump Status : OFF</li>
            <li>Tank Height : 21 CM</li>
        </ul>
        </div> ");
        } else if (distance > 9 && distance <= 11) {
client.println("70%;\"><span class=\"\">70% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div>
            </div>
            <ul>
                <li>Water Level :");
                    client.println(distance);
                    client.println("CM <span class=\"text-info\">MEDIUM</span>
</li>
                <li>Pump Status : OFF</li>
                <li>Tank Height : 21 CM</li>
            </ul>
            </div> ");
            } else if (distance > 11 && distance <= 13) {
client.println("60%;\"><span class=\"\">60% Full</span>");
                digitalWrite(ledPin, HIGH);
                client.println("</div>
                </div>
                <ul>
                    <li>Water Level :");
                        client.println(distance);
                        client.println("CM <span class=\"text-
info\">MEDIUM</span> </li>
                    </li>
                    <li>Pump Status : OFF</li>
                    <li>Tank Height : 21 CM</li>
                </ul>
                </div> ");
                } else if (distance > 13 && distance <= 15) {
client.println("50%;\"><span class=\"\">50% Full</span>");
                    digitalWrite(ledPin, HIGH);
                    client.println("</div>
                    </div>
```

```
                    <ul>
                        <li>Water Level :");
                            client.println(distance);
                            client.println("CM <span class=\"text-
info\">MEDIUM</span> </li>
                        </li>
                        <li>Pump Status : OFF</li>
                        <li>Tank Height : 21 CM</li>
                    </ul>
                    </div> ");
                    } else if (distance > 15 && distance <= 17) {
client.println("40%;\"><span class=\"\">40%
                            Full</span>");
                        digitalWrite(ledPin, HIGH);
                        client.println("</div>
                        </div>
                        <ul>
                            <li>Water Level :");
                                client.println(distance);
                                client.println("CM <span class=\"text-
info\">MEDIUM</span> </li>
                            </li>
                            <li>Pump Status : ON</li>
                            <li>Tank Height : 21 CM</li>
                        </ul>
                        </div> ");
                        } else if (distance > 17 && distance <= 19) {
client.println("30%;\"><span class=\"\">30%
                                Full</span>");
                            digitalWrite(ledPin, HIGH);
                            client.println("</div>
                            </div>
                            <ul>
                                <li>Water Level :");
                                    client.println(distance);
                                    client.println("CM <span class=\"text-
info\">MEDIUM</span> </li>
                                </li>
                                <li>Pump Status : ON</li>
                                <li>Tank Height : 21 CM</li>
                            </ul>
                            </div> ");
                            } else if (distance > 19 && distance < 21) {
client.println("20%;\"><span class=\"\">20%
                                    Full</span>");
                                digitalWrite(ledPin, HIGH);
                                client.println("</div>
                                </div>
```

```
                                <ul>
                                    <li>Water Level :");
                                        client.println(distance);
                                        client.println("CM <span class=\"text-
danger\">PARTIALLY EMPTY</span> </li>
                                    </li>
                                    <li>Pump Status : ON</li>
                                    <li>Tank Height : 21 CM</li>
                                </ul>
                                </div> ");
                                } else if (distance >= 21) {
                                client.println("10%;\"><span class=\"\">10%
Full</span>");

                                digitalWrite(ledPin, HIGH);
                                client.println("</div>
                                </div>
                                <ul>
                                    <li>Water Level :");
                                        client.println(distance);
                                        client.println("CM <span class=\"text-
danger\">PARTIALLY EMPTY</span> </li>
                                    </li>
                                    <li>Pump Status : ON</li>
                                    <li>Tank Height : 21 CM</li>
                                </ul>
                                </div>");
                                }


                                client.println("<div style=\"margin-top:
30px;\" class=\"card\">
                                    <div class=\"card-header bg-warning text-
light\"> Water Temperature | Humidity</div>
                                    <div class=\"card-body\">
                                        <h1> Temp: ");
                                            client.println(temperatureC);
                                            client.println("</h1>");
                                        client.println("<div class=\"card-
body\">

                                            <h1> Humidity: ");
                                                client.println(h);
                                                client.println("</h1>
                                        </div>");
                                        client.println("
                                    </div>
                                </div>
                                </div>
                                </div>");
                                client.println("
```

```
</body>

</html>");
```

3.7.2 ESP32 code in C

For implementing the code in ESP32, one must first set up the necessary environment which includes installing the Arduino IDE and the ESP32 board package. Then, connect the ESP32 board to the computer and select the appropriate board and port settings in the IDE. After that, i write the code in the IDE and upload it to the ESP32 board. Once the code is uploaded, the ESP32 will execute the code and perform the desired functions based on the instructions given in the code. Which is following given code for understanding.

```
// Load Wi-Fi library
#include <WiFi.h>
#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <DHT.h>

LiquidCrystal_I2C lcd (0x27, 16, 2);

#define echoPin 18 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 5   //attach pin D3 Arduino to pin Trig of HC-SR04

#define DHTPIN 19      // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 11
DHT dht (DHTPIN, DHTTYPE);

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable

int tank bar;
// Replace with your network credentials
const char* ssid = "project";
const char* password = "project1234";

const int ledPin = 15; // the number of the LED pin

// Set web server port number to 80
WiFiServer server (80);

// Variable to store the HTTP request
String header;
```

```
// Current time
unsigned long currentTime = millis ();
// Previous time
unsigned long previousTime = 0;
// Define timeout time in milliseconds (example: 2000ms = 2s)
const long timeoutTime = 2000;

const int oneWireBus = 4;
OneWire oneWire(oneWireBus);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

void setup () {
  Serial.begin(115200);

  lcd.init();
  lcd.begin(16, 2);
  lcd.backlight();
  dht.begin();

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT);

  pinMode(ledPin, OUTPUT);

  // Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();

  lcd.setCursor(0, 0);
  lcd.print("IP address");
  lcd.setCursor(0, 1);
  lcd.print(WiFi.localIP());
  delay(4000);
  lcd.setCursor(0, 0);
  lcd.print("SMART TANK LEVEL");
```

```
  lcd.setCursor(0, 1);
  lcd.print("MONITOR SYSTEM");
  delay(4000);
  lcd.clear();
}

void loop() {
  WiFiClient client = server.available();  // Listen for incoming clients

  float h = dht.readHumidity();

  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);
  float temperatureF = sensors.getTempFByIndex(0);

  lcd.setCursor(0, 0);
  lcd.print(temperatureC);
  lcd.setCursor(6, 0);
  lcd.print("H");
  lcd.print(h);

  lcd.setCursor(0, 1);
  if (distance <= 5) {
    lcd.print("PUMP STATUS: OFF");
  } else if (distance > 5 && distance <= 7) {
    lcd.print("PUMP STATUS: OFF");
  } else if (distance > 7 && distance <= 9) {
    lcd.print("PUMP STATUS: OFF");
  } else if (distance > 9 && distance <= 11) {
    lcd.print("PUMP STATUS: ON");
  } else if (distance > 11 && distance <= 13) {
    lcd.print("PUMP STATUS: ON");
  } else if (distance > 13 && distance <= 15) {
    lcd.print("PUMP STATUS: ON");
  } else if (distance > 15 && distance <= 17) {
    lcd.print("PUMP STATUS: ON ");
  } else if (distance > 17 && distance <= 19) {
    lcd.print("PUMP STATUS: ON ");
  } else if (distance > 19 && distance < 21) {
    lcd.print("PUMP STATUS: ON ");
  } else if (distance >= 21) {
    lcd.print("PUMP STATUS: ON ");
  }

  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;  // Speed of sound wave divided by 2 (go
and back)
  // Displays the distance on the Serial Monitor


  if (client) {  // If a new client connects,
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New
Client.");                                                      // print a message out
in the serial port
    String currentLine =
"";                                                             // make a String to hold
incoming data from the client
    while (client.connected() && currentTime - previousTime <= timeoutTime)
{  // loop while the client's connected
      currentTime = millis();
      if (client.available()) {  // if there's bytes to read from the client,
        char c = client.read();  // read a byte, then
        Serial.write(c);         // print it out the serial monitor
        header += c;
        if (c == '\n') {  // if the byte is a newline character
          // if the current line is blank, you got two newline characters in a
row.
          // that's the end of the client HTTP request, so send a response:
          if (currentLine.length() == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1
200 OK)
            // and a content-type so the client knows what's coming, then a
blank line:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            // Display the HTML web page
           client.println("<!DOCTYPE html>
            <html>
            <head>
            <title>Project</title>
            <link rel=\"stylesheet\"
href=\"https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.cs
s\"
```

```
             integrity=\"sha384-
Smlep5jCw/wG7hdkwQ/Z5nLIefveQRIY9nfy6xoR1uRYBtpZgI6339F5dgvm/e9B\"
crossorigin=\"anonymous\">
             <meta http-equiv=\"refresh\" content=\"1\">
             <style type=\"text/css\">
              .progress-bar-vertical { width: 300px; min-height: 400px;
display: flex; align-items: flex-end; margin-right: 20px; float: left; }
              .progress-bar-vertical .progress-bar { width: 100%; height: 0; -
webkit-transition:
              height 0.6s ease; -o-transition: height 0.6s ease; transition:
height 0.6s ease; }
               </style>
                </head>
                <body style=\"width: 100vw;height: 100vh;\"> <div
class=\"jumbotron bg-primary text-light\">
                 <div> <h1>Smart Tank Management System</h1>
                 <hr> dashboard </div>
                  </div>
                   <div class=\"container\">
                   <div class=\"card\">
                    <div class=\"card-header bg-primary text-light\"> Tank
Level </div>
                     <div class=\"card-body\"> <div class=\"progress progress-
bar-vertical\">
                     <div class=\"progress-bar progress-bar-success progress-
bar-striped active\" role=\"progressbar\"
                     aria-valuenow=\"100\" aria-valuemin=\"0\" aria-
valuemax=\"100\" style=\"height:");
if (distance <= 5) {
              digitalWrite(ledPin, LOW);
              client.println("100%;\"><span class=\"\">100% Full</span> ");
              client.println("</div> </div> <ul> <li>Water Level :");
              client.println(distance);
              client.println("CM <span class=\"text-success\">FULL</span>
</li> <li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
            } else if (distance > 5 && distance <= 7) {
              client.println("90%;\"><span class=\"\">90% Full</span>");
              digitalWrite(ledPin, LOW);
              client.println("</div> </div> <ul> <li>Water Level :");
              client.println(distance);
              client.println("CM <span class=\"text-success\">FULL</span></li>
<li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
            } else if (distance > 7 && distance <= 9) {
              client.println("80%;\"><span class=\"\">80% Full</span>");
              digitalWrite(ledPin, LOW);
              client.println("</div> </div> <ul> <li>Water Level :");
              client.println(distance);
```

```
            client.println("CM <span class=\"text-success\">FULL</span>
</li> <li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 9 && distance <= 11) {
            client.println("70%;\"><span class=\"\">70% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-info\">MEDIUM</span> </li>
<li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 11 && distance <= 13) {
            client.println("60%;\"><span class=\"\">60% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-info\">MEDIUM</span> </li>
</li> <li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 13 && distance <= 15) {
            client.println("50%;\"><span class=\"\">50% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-info\">MEDIUM</span> </li>
</li> <li>Pump Status : OFF</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 15 && distance <= 17) {
            client.println("40%;\"><span class=\"\">40% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-info\">MEDIUM</span> </li>
</li> <li>Pump Status : ON</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 17 && distance <= 19) {
            client.println("30%;\"><span class=\"\">30% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-info\">MEDIUM</span> </li>
</li> <li>Pump Status : ON</li> <li>Tank Height : 21 CM</li> </ul> </div> ");
          } else if (distance > 19 && distance < 21) {
            client.println("20%;\"><span class=\"\">20% Full</span>");
            digitalWrite(ledPin, HIGH);
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-danger\">PARTIALLY
EMPTY</span> </li> </li> <li>Pump Status : ON</li> <li>Tank Height : 21
CM</li> </ul> </div> ");
          } else if (distance >= 21) {
            client.println("10%;\"><span class=\"\">10% Full</span>");
            digitalWrite(ledPin, HIGH);
```

```
            client.println("</div> </div> <ul> <li>Water Level :");
            client.println(distance);
            client.println("CM <span class=\"text-danger\">PARTIALLY
EMPTY</span> </li> </li> <li>Pump Status : ON</li> <li>Tank Height : 21
CM</li> </ul> </div>");
          }


          client.println("<div style=\"margin-top: 30px;\" class=\"card\">
          <div class=\"card-header bg-warning text-light\"> Water
Temperature | Humidity</div>
           <div class=\"card-body\"> <h1> Temp: ");
          client.println(temperatureC);
          client.println("</h1>");
          client.println("<div class=\"card-body\"> <h1> Humidity: ");
          client.println(h);
          client.println("</h1> </div>");
          client.println("</div> </div></div> </div>");
          client.println("</body> </html>");
          // The HTTP response ends with another blank line
          client.println();
          // Break out of the while loop
          break;
        } else {  // if you got a newline, then clear currentLine
          currentLine = "";
        }
      } else if (c != '\r') {  // if you got anything else but a carriage
return character,
        currentLine += c;      // add it to the end of the currentLine
      }
    }
  }
  // Clear the header variable
  header = "";
  // Close the connection
  client.stop();
  Serial.println("Client disconnected.");
  Serial.println("");
 }
}
```

Chapter 4

RESULTS & TESTING

4.1. Results

The system was able to effectively monitor the water level in the overhead tank using the ultrasonic sensor and prevent overflow. The system was also able to automatically control the pump when the water level in the overhead tank was below a predetermined margin, thus preventing wastage of water and conserving electricity.

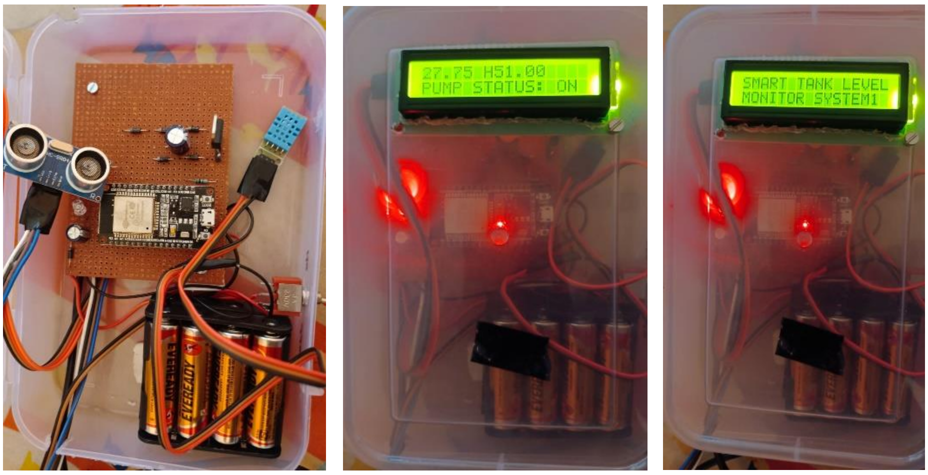

Figure 4.1

The web application was able to display all the important points such as the status of the pump, levels of water in the overhead tank, and status of the main tank. When the main tank was empty and the motor was off, the web application display unit automatically showed that the water level in the main tank was below the margin along with the status of the pump (OFF). This helped in preventing the overflowing of the water tank and controlling water wastage due to overflowing of water, while also saving electricity without overrunning the pump.

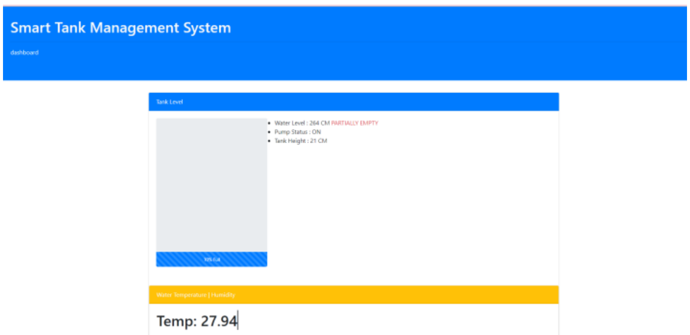

Figure 4.1 Designed web application

The ESP32 was able to effectively control the power supply of the pump, which was connected to the main circuit. The ESP32 Board was the control unit, which was connected with all the sensors and it controlled the setup as it was programmed. The sensors used in the system, including the Ultrasonic sensor, humidity sensor, and temperature sensor, were able to sense the distance, humidity, and temperature of the water in the tank, respectively. The ultrasonic sensor was able to detect the water level within a range of 7cm to 40 cm, and at a depth of 20 cm, the critical state buzzer would keep on buzzing between 20cm to 16 cm of water detection. At 7 cm depth, the ESP32 would switch off the motor considering the tank was full.

Overall, the implementation of the automated water tank and pump system using ESP32 was successful in effectively monitoring and controlling the water level in the tank, preventing water wastage and overflow, and conserving electricity.

4.2. Testing

The system was tested thoroughly to ensure its proper functioning. Different scenarios were tested to check the accuracy of the system's sensing and control mechanisms. Firstly, the system was tested to monitor the water level in the overhead tank. The ultrasonic sensor was found to be accurate in detecting the water level within the specified range of 7cm to 40cm. The buzzer and pump were activated correctly at the predetermined critical level and the water was pumped efficiently to the overhead tank.

Secondly, the system was tested to check the flow of water through the system. The flow meter was found to be working accurately and the ESP32 was able to track the flow data in real-time. The system was able to identify when the pump tank was empty and automatically turned off the pump to prevent any damage to the motor.

The temperature and humidity sensors were also tested to ensure that they were providing accurate readings. The temperature sensor was found to be working accurately and the system was able to detect changes in temperature in real-time. Similarly, the humidity sensor was able to detect changes in the humidity levels and identify the causes of water pollution.

Overall, the testing process confirmed the proper functioning of the system and its ability to prevent water wastage and conserve electricity.

Chapter 5

CONCLUSION

5.1 Conclusion

Water is a crucial resource for human survival, and it is used for a variety of purposes, including drinking, cooking, bathing, and more. Unfortunately, the water we rely on is increasingly becoming polluted due to various factors, such as industrial waste, agricultural runoff, and human activities. One of the main causes of water pollution is the overflowing of water tanks, which can lead to significant contamination of our water sources. To address this problem, we have developed a water level controller system that helps prevent water pollution and saves money by reducing water and electricity waste. Our system is designed to monitor the water level in tanks and prevent overflowing, thus reducing the amount of water that is wasted and preventing water pollution.

With our system, users can precisely regulate the amount of energy used, ensuring that no unnecessary water or electricity is consumed. This not only helps to save money in the long run, but also helps to conserve our natural resources and protect our environment. Our water level controllers are easy to install and use, and can be customized to meet the specific needs of different households and businesses. In addition to preventing water pollution, our system also helps to conserve energy and reduce electricity costs. By controlling the amount of energy used to pump water, our system helps to reduce unnecessary energy consumption, which in turn helps to reduce carbon emissions and protect the environment. It means our water level controller system is a powerful tool for protecting our water sources and conserving our natural resources. By preventing water pollution and reducing water and electricity waste, our system helps to create a sustainable future for all.

5.2 Future Scope

The future scope of this project is quite promising, and there are several directions that it can be taken in. Here are a few potential areas of development:

1. **Integration with smart homes:** The water level controller system can be integrated with other smart home devices, such as voice assistants or home automation systems.

This would allow homeowners to control the system through their voice, or set up automated schedules to run the system at specific times.

2. Mobile app integration: An accompanying mobile app can be developed to control the water level controller system remotely. This would allow users to monitor the water levels and control the system from anywhere, at any time.

3. Integration with water quality sensors: In addition to monitoring the water level, the system can also be integrated with water quality sensors to monitor the quality of water in real-time. This would allow homeowners to take action quickly if the water quality falls below safe levels.

4. Solar-powered water level controller: The system can be modified to run on solar power, which would make it eco-friendlier and more sustainable. This would be especially useful in areas with frequent power outages or limited access to electricity.

5. Integration with IoT: The system can be integrated with the Internet of Things (IoT) network, which would enable it to communicate with other devices and exchange data in real-time. This would allow for more efficient water management and better decision-making.

Overall, the water level controller system has a lot of potential for future development and can be customized to meet the needs of different users. As technology continues to advance, there are many possibilities for integrating the system with other devices and networks, and making it more user-friendly and sustainable.

6. References

1. Reddy, P. J. P., Viswanadh, G., & Singh, S. K. (2021, April). IOT based Smart Water Pump Switch. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 534-538). IEEE.
2. Sunmonu, R., Aduramo, S., Abdulai, O. S., & Agboola, E. A. (2017). Development of an Ultrasonic Sensor Based Water Level Indicator With Pump Switching Technique. *J. Electr. Eng*.
3. Getu, B. N., & Attia, H. A. (2016, December). Automatic water level sensor and controller system. In *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)* (pp. 1-4). IEEE.
4. Johari, A., Abd Wahab, M. H., Latif, N. S. A., Ayob, M. E., Ayob, M. I., Ayob, M. A., & Mohd, M. N. H. (2011). Tank water level monitoring system using GSM network. *International Journal of Computer Science and Information Technologies*, *2*(3), 1114-1120.
5. Sunmonu, R., Aduramo, S., Abdulai, O. S., & Agboola, E. A. (2017). Development of an Ultrasonic Sensor Based Water Level Indicator With Pump Switching Technique. *J. Electr. Eng*.
6. Wadekar, S., Vakare, V., Prajapati, R., Yadav, S., & Yadav, V. (2016, October). Smart water management using IOT. In *2016 5th International Conference on Wireless Networks and Embedded Systems (WECON)* (pp. 1-4). IEEE.
7. Kaur, D. (2016). IOT based Water Tank Control. *Article-Embedded for You] Jan/Feb*.
8. Vijayakumar, N., & Ramya, A. R. (2015, March). The real time monitoring of water quality in IoT environment. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-5). IEEE.
9. Maqbool, S., & Lone, S. A. (2015). GSM based Monitoring and Control of Systems.
10. Perumal, T., Sulaiman, M. N., & Leong, C. Y. (2015, October). Internet of Things (IoT) enabled water monitoring system. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)* (pp. 86-87). IEEE.
11. Z. Lin, W. Wang, H. Yin, S. Jiang, G. Jiao and J. Yu, "Design of Monitoring System for Rural Drinking Water Source Based on WSN," 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2017, pp. 289-293, doi: 10.1109/ICCNEA.2017.106.
12. M. Kumar Jha, R. Kumari Sah, M. S. Rashmitha, R. Sinha, B. Sujatha and K. V. Suma, "Smart Water Monitoring System for Real-Time Water Quality and Usage Monitoring," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 617-621, doi: 10.1109/ICIRCA.2018.8597179.
13. Mukta, M., Islam, S., Barman, S. D., Reza, A. W., & Khan, M. S. H. (2019, February). IoT based smart water quality monitoring system. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 669-673). IEEE.

14. M. B. Tephila, R. A. Sri, R. Abinaya, J. A. Lakshmi and V. Divya, "Automated Smart Irrigation System using IoT with Sensor Parameter," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 543-549, doi: 10.1109/ICEARS53579.2022.9751993.
15. J. -L. Chen, Q. -A. Wu and C. Ahmadi, "AIoT@Water - An Intelligent Water Resources Management System," 2022 5th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2022, pp. 199-204, doi: 10.1109/ICOIACT55506.2022.9971964.
16. H. Youness, G. Ahmed and B. E. Haddadi, "Machine Learning-based Smart Irrigation Monitoring System for Agriculture Applications Using Free and Low-Cost IoT Platform," 2022 International Conference on Microelectronics (ICM), Casablanca, Morocco, 2022, pp. 189-192, doi: 10.1109/ICM56065.2022.10005419.
17. K. B. R. Teja, M. Monika, C. Chandravathi and P. Kodali, "Smart Monitoring System for Pond Management and Automation in Aquaculture," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 204-208, doi: 10.1109/ICCSP48568.2020.9182187.

66666666

原創性報告

11%	7%	7%	2%
相似度指數	網際網絡來源	出版物	學生文稿

主要來源

1	M. S. Godwin Premi, Jyotirupa Malakar. "Automatic Water Tank Level and Pump Control System", 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019 出版物	2%
2	www.mdpi.com 網際網絡來源	1%
3	ijsret.com 網際網絡來源	1%
4	core.ac.uk 網際網絡來源	1%
5	www.ierjournal.org 網際網絡來源	<1%
6	www.researchgate.net 網際網絡來源	<1%
7	ebin.pub 網際網絡來源	<1%
8	Submitted to National American University 學生文稿	

<1%

9	Submitted to University of Ulster 學生文稿	<1%
10	Rahul Chauhan, Shubhanshu Kaintura, Niharika Dhyani, R. Gowri, Chandradeep Bhatt. "An IoT Enabled Smart Flask", 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), 2023 出版物	<1%
11	www.ajer.org 網際網絡來源	<1%
12	www.digitaljournal.com 網際網絡來源	<1%
13	Abdullah Al Shahid Chowdhury, Yasir Arafat, M. Shafiul Alam. "IoT-GSM Based Controlling and Monitoring System to Prevent Water Wastage, Water Leakage, and Pollution in the Water Supply", 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2022 出版物	<1%
14	dialnet.unirioja.es 網際網絡來源	<1%

15	Bam Bahadur Sinha, R. Dhanalakshmi. "Recent advancements and challenges of Internet of Things in smart agriculture: A survey", Future Generation Computer Systems, 2021 出版物	<1%
16	Submitted to University of East London 學生文稿	<1%
17	Submitted to Laureate Education Inc. 學生文稿	<1%
18	ejournal.raharja.ac.id 網際網絡來源	<1%
19	"Data Engineering and Communication Technology", Springer Science and Business Media LLC, 2020 出版物	<1%
20	Submitted to British University in Egypt 學生文稿	<1%
21	Submitted to University of Northampton 學生文稿	<1%
22	Submitted to University of Warwick 學生文稿	<1%
23	workingwithwater.filtsep.com 網際網絡來源	<1%

24	SJ. Madhumitha, S. Supriya, R Charanya, s Sudha.. "Survey on Automated Fluid Level Sensing and Controlling System Using IoT", 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020 出版物	<1 %
25	dokumen.pub 網際網絡來源	<1 %
26	www.ijrte.org 網際網絡來源	<1 %
27	Anagha Praveen, R. Radhika, Sidharth D, Sreehari Ambat, Anjali T. "Smart water level monitoring and management system using IoT", 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021 出版物	<1 %
28	Submitted to Liverpool John Moores University 學生文稿	<1 %
29	Sandhya.A. Kulkarni, Vishal D Raikar, B K Rahul, L V Rakshitha, K Sharanya, Vandana Jha. "Intelligent Water Level Monitoring System Using IoT", 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), 2020 出版物	<1 %

30	libguides.snhu.edu 網際網絡來源	<1 %
31	sciencepubco.com 網際網絡來源	<1 %
32	www.check-plagiarism.com 網際網絡來源	<1 %
33	"Cyber-physical Systems and Digital Twins", Springer Science and Business Media LLC, 2020 出版物	<1 %
34	"The Industrial Internet of Things (IIoT)", Wiley, 2022 出版物	<1 %
35	Peddamallu Jaya Prakash Reddy, Guntimadugu Viswanadh, Sanjay Kumar Singh. "IOT based Smart Water Pump Switch", 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021 出版物	<1 %
36	S Suraj, V Bharath, N.K Sridhar. "Wireless Automatic Water Level Controller", 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2018 出版物	<1 %

37	S. Swarnapali, P. Kumarage. "Real-Time Synchronized Monitoring of Water Consumption and Leak Detection in the Urban Water Supply of Sri Lanka", 2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI), 2020 出版物	<1 %
38	archives.univ-biskra.dz 網際網絡來源	<1 %
39	dspace.chitkara.edu.in 網際網絡來源	<1 %
40	gredos.usal.es 網際網絡來源	<1 %
41	imanagerpublications.com 網際網絡來源	<1 %
42	irep.iium.edu.my 網際網絡來源	<1 %
43	www.aeologic.com 網際網絡來源	<1 %
44	www.irjmets.com 網際網絡來源	<1 %
45	Érika Correia, Anderson Leite, Giovane Fernandes, João Vilasboas et al. "An Architecture for Social-Aware Navigation	<1 %

Based on a Chatbot Interaction", IOS Press, 2022
出版物

46 "Sustainable Communication Networks and Application", Springer Science and Business Media LLC, 2022 <1%
出版物

47 Submitted to Massey University <1%
學生文稿

48 Rajesh Singh, Mohammed Baz, Anita Gehlot, Mamoon Rashid et al. "Water Quality Monitoring and Management of Building Water Tank Using Industrial Internet of Things", Sustainability, 2021 <1%
出版物

49 "Industrial Internet of Things", Springer Nature, 2017 <1%
出版物

50 "Web, Artificial Intelligence and Network Applications", Springer Science and Business Media LLC, 2020 <1%
出版物

51 Farmanullah Jan, Nasro Min-Allah, Saqib Saeed, Sardar Zafar Iqbal, Rashad Ahmed. "IoT-Based Solutions to Monitor Water Level, Leakage, and Motor Control for Smart Water Tanks", Water, 2022 <1%
出版物

排除引述	開	排除相符處	關閉
排除參考書目	開		