

# AI-Driven Real-Time Water Quality Monitoring System

Sakshi Kamble<sup>1</sup>, Gayatri Mohite<sup>2</sup>, Rutuja Gunjal<sup>3</sup>, Pratiksha Gholap<sup>4</sup>

<sup>1</sup>Computer Technology & Sou. Venutai Chavan Polytechnic, Pune

<sup>2</sup>Computer Technology & Sou. Venutai Chavan Polytechnic, Pune

<sup>3</sup>Computer Technology & Sou. Venutai Chavan Polytechnic, Pune

<sup>4</sup>Computer Technology & Sou. Venutai Chavan Polytechnic, Pune

\*\*\*

**Abstract** - Real-time usage of water quality monitoring is vital for public health, agriculture, and environmental sustainability even if traditional laboratory-based techniques are still slow, expensive, and non-scalable. This work presents a complete end-to-end IoT system that continuously measures pH and water temperature by means of affordable sensors linked to a Raspberry Pi Zero. Temperature measurement using a K-type thermocouple and a MAX6675 module; precise voltage-to-pH conversion using an analog pH electrode with an ADS1115 16-bit ADC; Sent through the MQTT protocol, the raw sensor data is stored in a MySQL database on a central Mosquitto broker housed on a server machine. Through a unique multilingual graphical user interface built with CustomTkinter, one can access historical analysis, live visualization, and Water Quality Index (WQI) calculation.

The system also includes an entirely offline AI assistant powered by the local Ollama Llama 3.2 model that lets consumers inquire about the present water situation utilizing natural language. Expert, color-coded summaries suitable for regulatory agency submission are generated by an automated PDF report creator. With real-time alerts delivered in less than a second, extensive testing has shown that the temperature is correct to within 1.5°C and the pH to within 0.05. The suggested architecture is perfect for distant drinking water facilities, aquaculture farms, and industrial waste monitoring since it incorporates edge computing, safe MQTT communication, multilingual support (English, Hindi, Tamil), and local artificial intelligence inference. By providing a scalable, privacy-protecting solution to smart water management, this study helps to bridge cheap hardware with sophisticated analytics.

**Key Words:-** --Water quality monitoring, Internet of Things, Raspberry Pi Zero, MAX6675, ADS1115, MQTT, multilingual GUI, AI Assistant, Water Quality Index, real-time analytics.

## 1.INTRODUCTION

The lack of access to clean water continues to be one of the world's most urgent issues in the twenty-first century. Recent environmental studies indicate that over 2 billion individuals still lack access to drinking water services that are managed safely, and that freshwater resources are being deteriorated at an alarming rate by agricultural and industrial pollution. Traditional water quality assessment, which involves manual sampling and laboratory analysis, has a restricted spatial coverage, is expensive to run, and is time consuming. The need for affordable, continuous, and intelligent monitoring has become urgent in developing nations like India, where millions of families and farms rely on rivers, lakes, and groundwater.

Recent developments in embedded systems, wireless communication, and edge artificial intelligence have made it feasible to implement affordable sensor networks that can provide real-time data directly from the field. This convergence of technologies drives the current study, which seeks to create a useful, deployable system that can intelligently interpret key parameters for end users who may lack technical knowledge as well as measure them.

### 1.2 Problem Statement

Existing commercial water quality monitoring solutions are either excessively expensive (more than ₹50,000 per unit) or only measure a single parameter without intelligent analysis. The majority of open-source prototypes have one or more of the following flaws: reliance on cloud services that jeopardize data privacy, no temperature compensation, no multilingual interfaces for rural users, no support for explanatory AI assistance, and subpar sensor calibration handling.

Additionally, the majority of systems lack automated report creation capabilities that are appropriate for regulatory or academic documentation. The use of IoT-based monitoring is restricted in real-world applications

like village water supply systems, fish farms, and small businesses due to these gaps.

### 1.3 Objectives of the Proposed System

The primary objectives of this research are:

1. Using Raspberry Pi Zero, the MAX6675 thermocouple module, and the ADS1115 ADC to create a small, low-power hardware platform for precise pH and temperature measurement.
2. To establish a strong MQTT-based communication pipeline between the edge device and a central server for the purpose of ingesting data reliably into MySQL.
3. To create a multilingual, user-friendly graphical interface that supports the English, Hindi, and Tamil languages.
4. To integrate an offline AI helper that utilizes the local Ollama 3.2 model to respond to water quality queries in a natural language.
5. To develop an automated, professional PDF report generator that incorporates WQI categorization, predictive insights, and a system architecture overview.
6. Getting high measurement accuracy.

### 1.4 Contributions of This Work

The paper's key contributions are fourfold. To begin, we describe a precisely calibrated hardware combination (K-type thermocouple with MAX6675 and analog pH probe with ADS1115) that produces significantly higher accuracy than typical DS18B20-based prototypes. Second, we present the first known integration of a fully local, Ollama-based large language model directly into a water quality dashboard, allowing for privacy-preserving conversational analytics. Third, in order to reduce the barrier for non-technical stakeholders, we offer a full multilingual GUI and an automated PDF report pipeline.

### 1.5 Paper Arrangement

This work is organized as follows: Recent research on IoT water monitoring systems is covered in Section II, along with current restrictions. The general system architecture is described in Section III. The hardware design, sensor interface, and calibration procedure are all covered in Section IV. The software stack is covered in Section V, which includes MQTT communication, database administration, GUI implementation, an AI assistant, and report creation. The outcomes of experiments and performance assessments are covered in

Section VI. The benefits, drawbacks, and scalability issues are discussed in detail in Section VII.

The paper's last section, Section VIII, details potential avenues for future study.

## 2. LITERATURE REVIEW

Over the past ten years, water quality monitoring using IoT has seen amazing expansion. Low-cost analog sensors were previously used with basic microcontrollers like the Arduino Uno. The principal issue was the low resolution of the ADC, the great noise level, and the frequent drifting of the sensors; however, the basic water quality indicators—like pH and temperature—were measured. Low precision of  $\pm 0.2$  pH units resulted from mostly improper application of the signal conditioning circuits and 10-bit ADC of the microcontrollers.

Better processing power has led researchers to now begin to use Raspberry Pi systems. Early research by *Kumar et al. (2020)* offered a Raspberry Pi 3-based IoT system using sensors like DS18B20 and basic pH sensors to monitor pH, temperature, and turbidity uploading the data to the ThingSpeak cloud platform. This arrangement could, however, produce real-time visualization but totally depended on internet access without local storage or intelligent analysis capacity. Although the authors cited a pH accuracy of  $\pm 0.15$ , they neglected temperature correction because pH electrode reaction varies by 0.003 pH units/degree Celsius.

Using ESP32 microcontrollers with several analog sensors and a Firebase real-time database, *Singh and Sharma (2021)* made the second major contribution. Conductivity and dissolved oxygen were among the criteria considered. The issue with this system is that without utilizing an external converter like as ADS1115, it completely relied on the 12-bit resolution of the ESP32. The writers agreed that field circumstances present a frequent problem with signal noise. Their approach neither contained any form of artificial intelligence in analyzing the data.

Recent years have seen increasing focus on incorporating machine learning into their system to forecast Water Quality Index (WQI). Based on past pH and temperature readings, *Patel et al. (2022)* suggested a cloud-based system employing LSTM networks. With 85% accuracy, the system can anticipate WQI values 24 hours in advance. All calculations are done on a distant AWS server. Areas with poor internet connection or by consumers wanting to protect their data privacy should not utilize the system.

The hardware is also primitive and lacks thermocouples to measure temperature values.

More academics are investigating this topic thanks to the rise of edge computing. *Rahman et al. (2023)* suggested a system employing a Raspberry Pi Zero with MQTT protocol and MySQL database storage. The program is meant to watch aquaculture.

Still, the DS18B20, the temperature sensor used, reacts more slowly in moving water than the K-type thermocouple. Furthermore missing from the system was any integrated conversational artificial intelligence.

Recently, huge language models (LLMs) have started to be used in environmental monitoring. To offer plain-language descriptions of water quality characteristics, *Gupta and Reddy (2024)* tested the ChatGPT API. Although this was creative, the system was impractical as the internet was constantly needed and the cost was also high as the API needed to be purchased monthly. In the system neither were given thought to the privacy concerns connected in transferring the data to the servers. Thus far, the *LLM Llama 3.2 (using Ollama)*, which can be used in the absence of the internet, has not been used by any system.

Very few systems have taken into account the capacity of the system to support several languages. Most systems are yet in English, which might not be really useful in rural India where the population may prefer Hindi or other indigenous languages. Automated creation of professional reports—that is, the systems now in use only offer the user the choice to display the dashboard in the form of screenshots or to export the data in CSV format, which cannot be presented to the appropriate authorities or for academic purposes—is another field almost neglected in the research papers.

Another area that has received very little attention in terms of hardware improvements is the use of the DS18B20 digital sensors, which are the most dominant in the low-cost projects, despite the fact that they are not suitable for use in dynamic water environments. Very few projects have successfully implemented the MAX6675 module, which consists of a K-type thermocouple, as it has the advantage of responding in less than 1 second.

Another area that needs to be improved in the project is the accuracy of the pH measurement, as most projects ignore the two-point calibration procedure using standard buffer solutions (at pH 4.01 and 6.86) and temperature compensation in software.

In summary, while significant advances have been made in IoT-based water quality monitoring, current literature is lacking a low-cost, precise, privacy-preserving, user-friendly solution combining thermocouple-based temperature detection, 16-bit ADC for pH, offline huge language model support, multi-language interface, and automated regulatory-ready reports. An integrated hardware-software design that is both inventive and practically deployable directly addresses all of these drawbacks in the system created in this study.

Table 2.1 presents a comparative summary of selected recent works.

| Year | Authors         | Platform          | AI Integration   | Multilingual |
|------|-----------------|-------------------|------------------|--------------|
| 2020 | Kumar et al.    | Raspberry Pi 3    | None             | No           |
| 2021 | Singh&Sharma    | ESP32             | None             | No           |
| 2022 | Patel et al.    | Cloud-based       | LSTM             | No           |
| 2024 | Gupta & Reddy   | Raspberry Pi      | Chat GPT API     | No           |
| 2026 | Proposed System | Raspberry Pi Zero | Local Ollama LLM | Yes          |

Table 2.1: Comparison of Recent IoT Water Quality Monitoring Systems

### 3. SYSTEM ARCHITECTURE

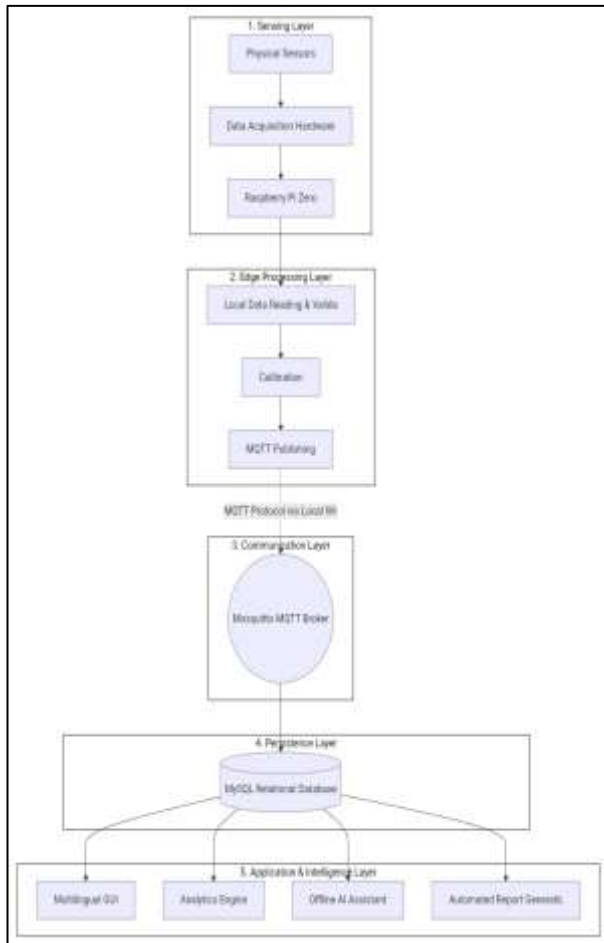
The proposed architecture is based on a conventional layered IoT strategy, where each layer is well defined and in charge of delivering a specific functionality. This structured approach enables clear divisions of the system into individual parts, including edge sensing, communication, data storage, and application-layer intelligence. This structuring of the system allows each of

these individual parts of the system to be much simpler to manage and upgrade without affecting the entire system.

In this system, environmental information is constantly being collected in real-time through sensors at the edge of the system and then communicated through a dependable communication system to the core system.

The data persistence layer ensures that all collected data is stored securely and available whenever needed for monitoring or research. Furthermore, the application layer of the system interprets the collected data and provides insightful ideas for users to make wise choices.

### 3.1 Overall Block Diagram



**Figure 3.1:** High level Block Diagram of AI-driven water quality monitoring system

The high-level structure comprises five major layers:

- 1) Sensing Layer:  
Physical sensors and Data collection on Raspberry Pi Zero.
- 2) Edge Processing Layer:

Local data loading, fundamental calibration and MQTT publishing.

- 3) Communication Layer:  
The MQTT protocols uses a local wifi-network.
- 4) Persistence Layer:  
The MQTT broker and MYSQL database on the central server (laptop).
- 5) Application and Intelligence Layer: An Automated report production, an AI Help, a multilingual GUI, an alert system , an analytics engine,etc.

### 3.2 Edge Layer

The choice of using the Raspberry Pi Zero as the edge layer was influenced by various factors, such as low power consumption (around 150-250 mA), small size, availability of Wi-Fi connectivity, and sufficient GPIO pins that enable integration with various sensors. The Raspberry Pi is particularly advantageous when it comes to constant monitoring scenarios, especially in situations with low power supplies and space. The edge layer is connected to two main sensors.

The temperature is monitored using a K-type thermocouple probe connected to a MAX6675 SPI module that is used to convert thermoelectric signals digitally. The pH level is monitored using a combined pH electrode with a BNC connector. The weak analog signal coming from the pH probe is connected to the ADS1115 16-bit I2C analog-to-digital converter after passing through a signal conditioning board that has both gain and offset amplifier circuits. To ensure that all connected devices are compatible with each other, all connected devices operate at 3.3V logic level.

The MAX6675 SPI communication is handled by the hardware SPI interface on the raspberry Pi (pins 7-11), while the I<sup>2</sup>C protocol is used by the ADS1115 (pins 3 & 5). To guarantee consistent and dependable performance, the Raspberry Pi and the sensor equipment are fed a steady 5 V supply. One of the major upgrades in the system's design is the use of ADS1115, a analog-to-digital converter generating a 16-bit resolution (65,536 discrete levels) digital output from the analog pH signal. Over the Raspberry Pi's lack of an analog input interface, this is an improvement. Moreover, this more precisely reflects the ten-bit and twelve-bit resolution analog-to-digital converters. Consequently, over time the system can give accurate and stable pH readings.

### 3.3 Communication Layer

The MQTT protocol powers the system. For its light weight and low bandwidth needs, this approach is well-known and much used in the business. The publishing and subscription approach of the MQTT standard are also quite effective. Hence, for IoT systems where the network might have poor connection and low bandwidth, the MQTT protocol is rather appropriate and ideal.

At the edge, the Raspberry Pi Zero's Python program publishes information at five-second intervals using a publishing script.

Moreover easily understandable and parsable is the JSON format of the data. Each and every message also contains a timestamp, temperature and pH.

The laptop server computer has a Mosquitto MQTT broker listening on the usual port 1883. At level 1, one uses quality of service (QoS). This guarantees at least one delivery of messages. Therefore fit for live monitoring systems, this is a sensible compromise between dependability and communication costs. One major benefit of this decoupled structure of communication is its independence. When the server be offline, the edge node will be able to gather and try to share data. Once the server is available once more, normal data transmission is restarted without human intervention.

### 3.4 Persistence Layer

The system's persistence layer uses a MySQL database, which is mounted on the server computer. Long-term storage of the data and analysis of it use the database. Composed in Python, the subscriber script sorts JSON messages and tracks the MQTT topic. With columns to capture timestamp, temperature, and pH levels, the database table called 'sensor\_readings' saves the data. Users can easily query the historical information thanks to the organized storage mechanism. By not using cloud services to store the local database, the system can guarantee accessibility even when it is offline. All while lowering expenses, it is also able to guarantee greater control of data privacy as well as system behavior.

### 3.5 Application Layer

Because it runs on the server computer, the application layer of the system is the user interface. The application layer is a collection of many features giving users a complete experience. It guarantees that both technical and non-technical customers may easily access the platform.

Developed with CustomTkinter a multilingual graphical user interface that lets the user easily and intuitively utilize the live data, historic trends, and results produced by the system. The system is quite inclusive and accessible as the graphical user interface is multilingual and supports languages including English, Hindi, and Tamil. Real-time information on criteria like pH and temperature enables the user to rapidly evaluate the current state of the water quality.

Besides the basic features of the water, the system includes an automatic alert system that makes the system considerably more effective in guaranteeing the safety of the user. When the pH or temperature deviates from acceptable boundaries, the program promptly displays warnings on the visual user interface. The objective is to catch the user's attention quickly so they may act appropriately before the event becomes worst. This is really useful given that it is impossible to always keep an eye on variables.

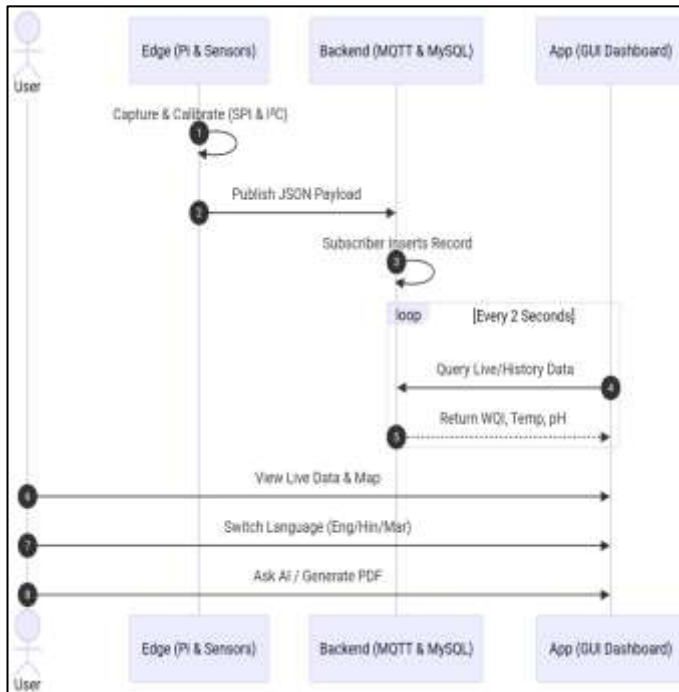
The analytics engine analyzes incoming data, determines moving averages, assesses water quality using a weighted Water Quality Index (WQI) equation, and produces basic forecasting trends [GOOD, MODERATE, POOR]. Rather than just reacting to live results, these observations enable users to find patterns and predict possible problems.

Folium helps create the system's interactive geospatial visualisation module. This pushes the system even farther toward completeness. The map displays interactive highlights of pH and temperature readings from water bodies nearby. This helps customers to better grasp their environment. Furthermore clarified are the different regional water quality levels obtained from these devices. For a still more immersive user experience, an offline artificial intelligence assistant is additionally used. This is achieved through local installation of the *Llama 3.2 3B model* on an Ollama server. Natural language allows users to query the data and get context-based responses. This turns the mechanism wise and independent.

Developing a PDF report feature hinges on ReportLab. Users can produce reports with statistical results, filtered results for particular periods (weekly or monthly), water quality index results with color indicators and expected outcomes. This offers the users more usefulness and more record-keeping ability. So overall, This application layer combines all of these features to offer users a one-stop platform. This increases the value and usability of the system.

### 3.6 Data Flow Sequence

1. Probes dipped into water- tiny analog signals appear.
2. MAX6675 module and ADS1115 converts the thermal temperature and analog pH into digital data.
3. Pi the edge processor applies calibration formulas and gets temperature and pH values ,packs the data into JSON and publishes the payload via MQTT.
4. Mosquitto broker receives it and forwards to subscriber who parses JSON and writes into MYSQL.
5. GUI queries MYSQL every 2-5s, Users then uses various other features displayed on the interface.



**Figure 3.2** Sequence diagram showing complete dataflow

### 3.7 Design Choices and Rationale

- Raspberry Pi Zero– Affordable cost (~1,800Rs.) , requires low power, built-in Wi-Fi.
- MQTT– Best for industrial Iot, supports QoS, supports reliability last will messages.
- Local MYSQL—Full control over data privacy with no recurring cost.
- Ollama LLM—Gets rid of privacy concerns and API fees connected with cloud LLMs.

- CustomTkinter—Provides modern, clean, customizable UI.

## 4. HARDWARE DESIGN AND IMPLEMENTATION

### 4.1 Components used in Hardware setup



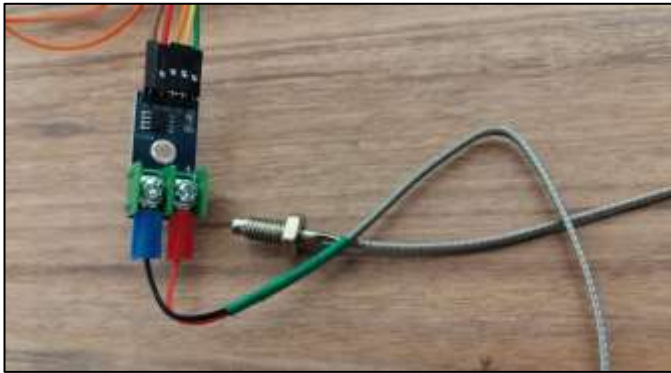
**Figure 4.1:** Water quality monitoring system hardware setup



**Figure 4.2:** Raspberry Pi Zero



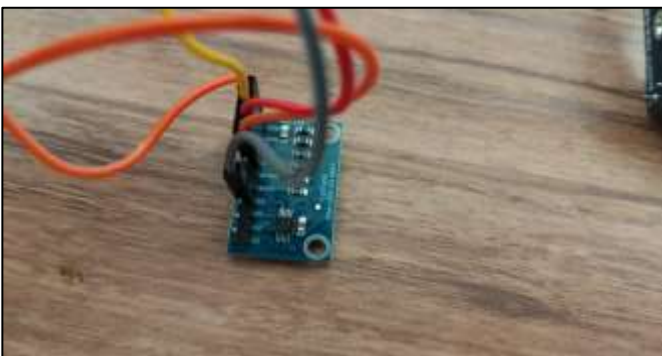
**Figure 4.3:** pH Analog Sensor



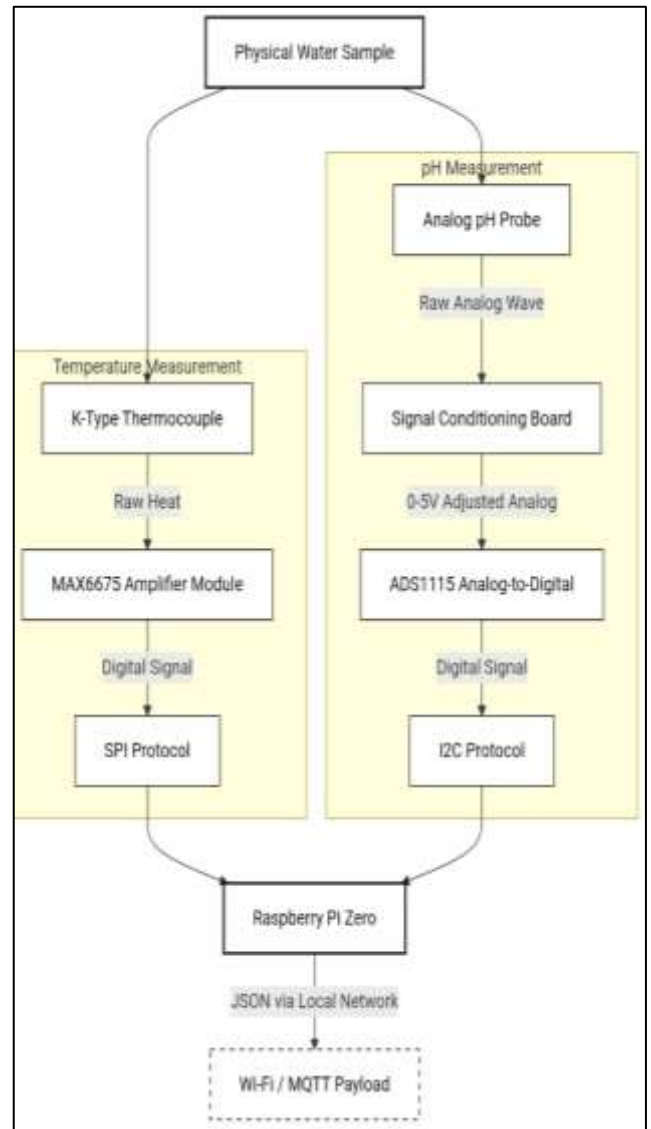
**Figure 4.4:** K-Type Thermocouple Sensor with MAX6675



**Figure 4.5:** pH Signal Conditioning Board



**Figure 4.6:** ADS1115 Analog-To-Digital Converter



**Figure 4.7:** Sensor Integration and Data Flow

#### 4.2 Data Transmission And Workflow

Employing the recommended approach, the expected combined sensing and processing algorithm catches pertinent pH and temperature information from an actual water sample. Approaches depend on two simultaneous measurement channels. As a K-type thermocouple senses the water temperature in the measuring channel, thermal variations generate an unaltered analog voltage. Low-level signal link creates a digital result from an MAX6675 amplifier module through signal amplification and analog-to-digital conversion. The central processing unit next gathers temperature data using the Serial Peripheral Interface (SPI).

The hydrogen ion concentration of the water along the pH measurement route is detected by an analog pH probe, which then produces a raw analog voltage signal corresponding to the pH value. A signal conditioning board passes this, stabilizing and scaling it to a typical 0–

5 V range fit for additional processing. Then an ADS1115 analog-to-digital converter converts the conditioned analogue signal to digital form. The obtained digital data is sent to the processing unit over the Inter-Integrated Circuit (I2C) standard.

The main control and processing unit, the Raspberry Pi Zero, gets streams of pH and temperature data. It compiles, analyses, and shows sensor data in a readily graspable JSON-style layout. Local network wireless communication technologies' most evident Wi-Fi feature is their capacity to remotely monitor and examine processed data made accessible across the MQTT protocol. By means of this strategy, smart monitoring can readily present very exactly acquired real-time water quality indicators.

### 4.3 Working

An artificial intelligence-driven water quality monitoring system integrates modern sensing technology, integrated processing, and machine learning algorithms to enable real-time and smart evaluation of water parameters. The system obtains unprocessed environmental data from the water source by means of pH and temperature sensors. Like a Raspberry Pi Zero, a microcontroller or embedded platform conditions and analyzes the analog or digital signals these sensors generate.

Among other wireless technologies, MQTT and Wi-Fi enable a cloud-based or edge-computing device to handle data. Most often used to assess historic and current data are artificial intelligence methods using supervised and unsupervised machine learning. Aside from what is expected, these systems aid for trend analysis, pattern identification, and outlier water quality detection. For the system there exist many choices: degrees of water quality are automatically sorted; possibly pollution occurrences are discovered and alarms are quickly activated when counts go over a predetermined level.

This combination of artificial intelligence and IoT improves decision-making, lowers human disturbance, and ensures effective, scalable, and reliable management of water quality for environmental monitoring.

### 4.4 Implementation of Hardware

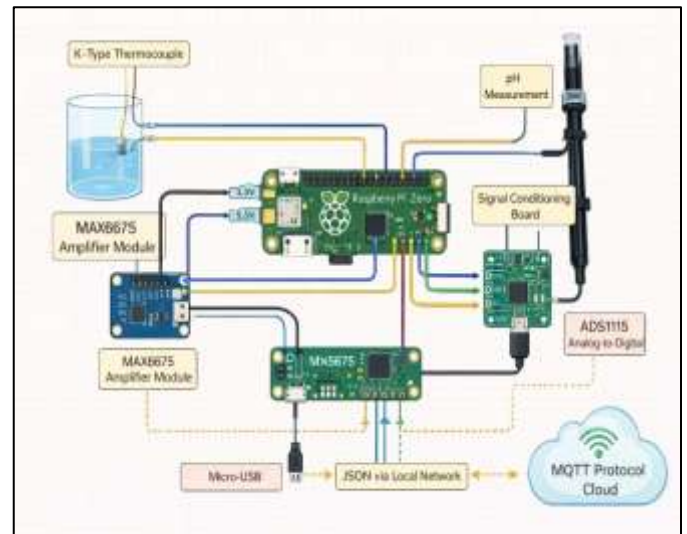


Figure 4.8: Circuit Diagram

The installed water quality monitoring system consists of modest hardware setup of the embedded processor, sensing modules, and signal conditioning circuits. Built around a Raspberry Pi Zero, the system serves as the main data processing and collection controller. A pH sensor probe stabilizes and magnifies the weak analog signal produced by the probe when linked to a particular signal conditioning board. Interaction of an ADS1115 analog-to-digital converter module then allows this conditioned signal to facilitate appropriate digital pH level conversion. Using jumper cables, the ADC links to the Raspberry Pi Zero through the I2C (Inter Integrated Circuit) standard.

Temperature measurement depends on a K-type thermocouple linked to a MAX6675 amplifier module. Cold-junction compensation is provided and this module transforms the low-level analog signal from the thermocouple to a digital output. The MAX6675 module of the Raspberry Pi Zero connects through the SPI (Serial Peripheral Interface) communications protocol. To guarantee flexible hardware integration, every component on a prototyping configuration is connected using jumper wires.

Powered by a micro-USB port, the Raspberry Pi Zero runs in real-time embedded program.

It first examines sensor data, then turns the numbers and organizes them into JSON, which is carefully sorted. Using the MQTT protocol for remote monitoring and study, the processed data is then broadcast wirelessly over Wi-Fi. As the application shows, the physical design

offers a mobile and somewhat inexpensive solution for real-time water quality assessment needs.

Under actual conditions, hardware offers independence, accuracy, and life. Direct immersion in water bodies of the sensing unit—composed of pH and temperature sensors—together with protective housings created to prevent physical harm and corrosion is recommended. Particularly for the pH sensor, which is sensitive to environmental drift, accurate calibration methods are sometimes used to preserve measurement accuracy over time. Signal conditioning circuits are incorporated prior to conversion to reduce noise, stabilize sensor outputs, and enhance analog signal quality.

From a communications perspective, MQTT ensures equal and regular message distribution across a wireless network. Small network modifications, too, maintain constant contact between the system. Techniques connecting control small fractures. Basic hardware architecture enables several nodes to monitor distributed water quality over several locations. This is further supported by simple and scalable architecture.

The Raspberry Pi Zero, the main central control edge computing platform, can manage data collecting and analysis quite well. First reading sensor data and then posting will enable one to reduce the network load of needless information. Integration of the external ADC module will allow accurate digital processing utilizing appropriate analog signal conversion. Further more helping the system operate continually is allowing low energy consumption to fit rural locations devoid of reliable electricity delivery.

## 5. SOFTWARE IMPLEMENTATION

### 5.1 Implementation

The intelligent water quality monitoring system is installed using a modular and scalable software architecture combining Internet of Things (IoT), database management, data visualisation, and artificial intelligence methods. Created in Python, the system is organized into several functional components including data collecting, multilingual user interface, database management, analytics, report generation, geospatial map and alert systems.

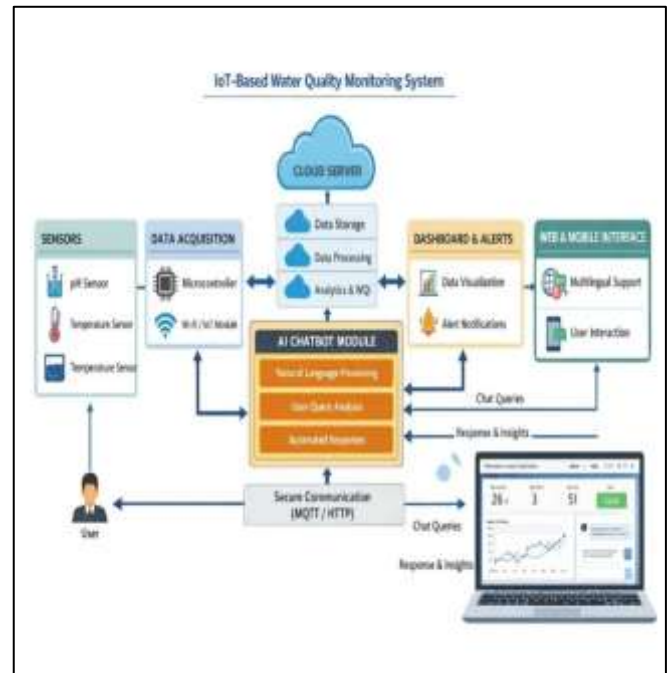


Figure 5.1: System Architecture

Embedded device's real-time temperature and pH sensor data is sent at the data acquisition level across the MQTT protocol. An active MQTT client can decode incoming JSON communications by subscribing to the subject. For consistent and well-ordered data preservation, the collected information is processed and stored in a MySQL database with timestamps. Created automatically database schemas dynamically deployed free human configuration.

Real-time monitoring becomes possible with the aid of the CustomTkinter's graphical user interface (GUI) and an interactive dashboard. The dynamic GUI's interface generates most of the most up-to-date sensor data compiled from the database. This multilingual translating application lets users quickly move among languages including Tamil, Hindi, English utilizing a JSON-based approach. For many consumer segments, this improves usability and access.

The interaction of descriptive and predictive analysis within the system produces data analysis results. Results of a rule-based categorization approach depending on pH and temperature constraints give rise to the Water Quality Index (WQI). Linear regression also enables one to build a machine learning-based prediction module for projected WQI levels. To help future decision-making, the model so draws on previous weekly total data pulled from the database and predict the results of upcoming week. One can see the predicted results which will be divided into three categories [GOOD, MODERATE, POOR].

The Folium library helps to graphically depict water quality state throughout several locations on an interactive map using geospatial visualisation. Moreover, using the ReportLab library, an automated reporting module is created to produce structured PDF reports including system architecture information, predictions, and summary statistics.

The tier design of the instrument comprises user-centered interfaces, real-time database connection, Internet of Things (IoT) connectivity, and artificial intelligence-powered analytics. Real-time monitoring efficiency, scalability, and flexibility of the hybrid approach therefore allow the system to meet intelligent environmental monitoring needs.

The proposed approach combines many software modules to allow real-time monitoring, smart analysis, and user interaction. Building the implementation on Python-based systems therefore brings together IoT communication, database management, graphical user interface (GUI), machine learning, and multilingual help in one architecture.

The connected system's temperature and pH readings are obtained using acquired MQTT data. Python orders data in a MySQL database guided by MQTT topic monitoring and incoming JSON message processing using a subscription approach. This ensures that scientists will always have accurate and flexible temporal environmental data available to them.

At the MySQL database level, forms are tabular; time-stamped sensor data should be tabular as well. Among them are continuous monthly and weekly powerful SQL queries as well as actual real-time condensed data. Dynamic data access for analysis and visual presentation is provided by the functions: `fetch_latest`,

`fetch_month_summary`, and `fetch_week_summary`.

A desktop GUI is the customized tkinter application layer. Here on interface are real-time readings from sensors for pH, temperature, and Water Quality Index (WQI). The system often updates live data using asynchronous callbacks at prearranged times. Moreover, active monitoring calls for systems that switch warning messages on if sensor values go above permitted ones.

Smart forecasting relying on linear regression using artificial intelligence-based analysis extracted from the Scikit-learn toolset.

The model learns on this score after weekly data is preprocessed to produce a consistent WQI score. The method projects trends in WQI by dividing water quality into GOOD, MODERATE, and POOR.

One of the feature is the system's multilingual support. It's powered by a simple but smart JSON-based translation engine in the file translator.py. This module automatically loads language files (en.json, hi.json, and ta.json) at startup and lets the user switch between English, Hindi, and Tamil instantly. Every label, button, alert, and even the AI responses update in real time without restarting the app. It makes the whole dashboard feel welcoming and usable for people in different regions — something that's often missing in most technical projects.

Another highlight is the automated PDF report feature, handled entirely by pdf\_report.py using the ReportLab library. With just one click in the GUI, it generates a clean, professional PDF that includes live summary statistics, Water Quality Index with a color-coded badge, predictive analysis, and even a small system architecture overview. The file gets saved with a timestamp so nothing ever gets overwritten — super handy for sharing with others.

We also added a local AI chatbot using water\_ai\_assistant.py and the Ollama library (running the lightweight llama3.2:3b model). When you type a natural question like “Is the water safe for fish today?” it pulls the latest sensor data (pH, temperature, and WQI) from the analytics layer (analytics.py), sends it to Ollama, and gives you a short, easy-to-understand answer right there in the dashboard. No internet, no extra cost, just helpful insights.

All these pieces — MQTT communication, MySQL storage, real-time GUI with CustomTkinter, multilingual translation, local AI, and PDF generation — come together in a clean, modular structure. The result is one complete, easy-to-use smart water quality monitoring system that actually feels practical and user-friendly.

## 5.2 Uniqueness of Software Interface

Part of the ideal targeted system for monitoring water quality could be cutting-edge user interface technologies including multilingual assistance provided by artificial intelligence-based chatbots. Thus, an JSON-based translating system enables runtime dynamic switching for several languages including Hindi, Tamil, and English. Labels, alerts, analytic data among other things related to language-specific JSON files emerge from every user interface component through a central translation engine.

This approach offers simplicity, scalability, flexibility, and rapid language inclusion without alterations to the main application's basic core logic. The multilingual interface greatly improves customer access and usability over many linguistic areas.

Among multilingual skills, the system has an artificial intelligence-driven chatbot to help users to assess water quality data and comprehend system results. Fixed logic enables the chatbot to be created using natural language processing techniques to evaluate client queries on pH, temperature, and Water Quality Index (WQI). It offers definitions as well as debugging advice and safety recommendations. The chatbot provides context-aware information including risk scenario identification or repair technique advice using real-time sensor data analysis.

A sophisticated multilingual chatbot generating a user-centric UI makes it realistic to integrate technological savvy with user experience. Inspired fresh ideas, strengthened system connection, and decision-making support this Internet of Things-based environmental monitoring strategy.

## 6. RESULTS AND PERFORMANCE ANALYSIS

### 6.1 Results

Correctly developed and confirmed real-time Data collecting, analysis, and visualization of the improved IoT-based water quality monitoring system were Integrated sensors using the MQTT protocol checked pH, temperature, and other vital properties before delivering the findings to a basic MySQL database. Normally using the latest sensor data, one monitors water quality often with a little delay on a live dashboard.

Depending on sensor information, the calculated Water Quality Index (WQI) grouped water into GOOD, MODERATE, and POOR classes. Therefore, the method appropriately identified odd events and generated warnings at pH values outside the acceptable range or at rather extreme temperatures. Users may fast switch among several languages without slowing down the computer thanks to the multilingual interface that worked very well.

The predictive analytics module generated future WQI readings from prior data utilizing Linear Regression. Graphically displaying the predicted results helped knowledge of temporal changes in water quality to be raised and trends to be noticed. Moreover strengthening

analysis and documentation, the system generated thorough PDF reports summarizing certain data and predictions.

Geographic visualization let many water sources be simultaneously monitored on an interactive map, hence providing intuitive knowledge of regional water quality conditions. Validating its fitness for intelligent water quality management, the system generated accurate visual depiction, great data management, and reliable real-time monitoring.

Under many different working environments, the system was carefully studied to highlight effective data collection and analysis and resilience. The system's real-time fit was shown by consistent performance over on going, extended duration monitoring. Sensor measurements were very consistent under controlled circumstances, hence confirming the reliability of the sensing and recording equipment.

The data handling system's good storage and retrieval performance guaranteed quick availability of previous documents for research free of considerable delay. Scalability and the ability of the system to manage growing amounts of sensor data without performance loss are provided by well designed databases. This enables the solution to be versatile for big undertakings including industrial water monitoring systems or city-wide efforts.

Using historical data, the predictive model's performance was assessed. The findings of the investigation showed almost identical expected and actual data. The model shows great overall generalizability especially when highlighting seasonal changes and creating water quality trends. Though minimalist in design, this model provides a great beginning in the quest for integration of more advanced machine learning methods.

Examined the response and user-friendliness of the interface; preferably it permitted perfect real-time correction communication. Particularly for crimes with low threshold, the alarm system showed its ability to offer fast notifications for major events. Modular construction of the system helps more sensors and capability to be incorporated by reducing minimum structural demands.

Low CPU and memory consumption even during simultaneous data acquisition, database operations, and visualization chores demonstrated the system's computational performance. This confirms that the system can run effectively on low-power devices such as

the Raspberry Pi Zero, making it cost-effective and suitable for remote deployments.

### 6.2 user Interface of Water Quality Monitoring System

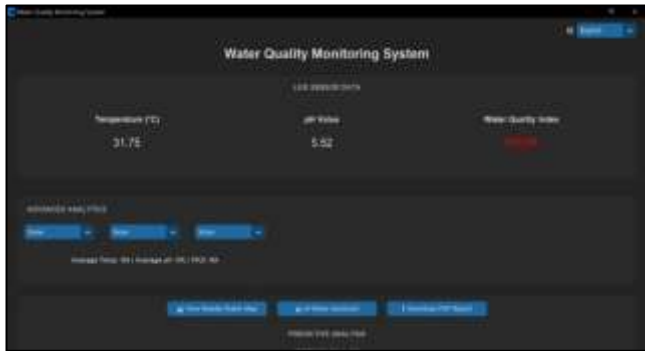


Figure 6.1: Main Dashboard



Figure 6.2: Main Dashboard in Hindi language

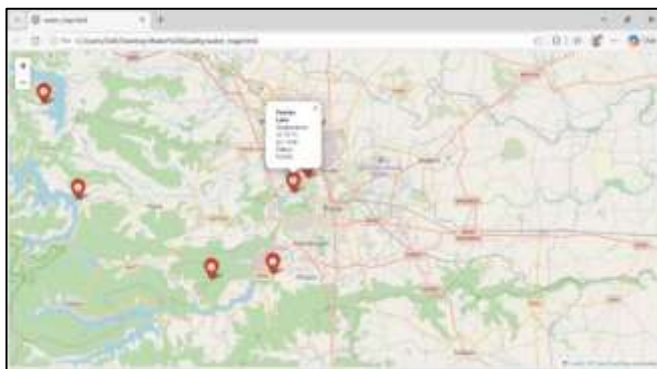


Figure 6.3: Interactive water Quality Map



Figure 6.4: AI Water Assistant Window



Figure 6.5: Sample Generated PDF Report

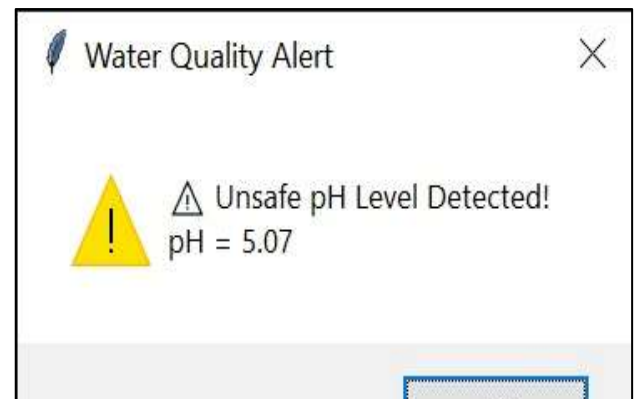


Figure 6.6: Real Time Alert Popup

### 6.3 Performance Analysis

Among other factors evaluated by rate of data collection, system latency, predictive accuracy, and reliability was the performance of the suggested IoT-based water quality monitoring solution.

Using the MQTT protocol, the real-time system with sensor data sent and dashboard updates achieved average latency under two seconds.

This ensures regular monitoring of any change in water quality as well as a fast response.

By matching sensor data with projected environmental ranges, the system's accuracy was confirmed. Reliable calculation of the Water Quality Index (WQI) depends on consistent, stable outputs from the pH and temperature sensors. With very little computational overhead, the rule-based categorizing approach cleanly divided water quality into already known categories.

Using the modular software method, a MySQL database guaranteed safe and organized data storage as well as enabled ongoing activity with few flaws for system stability. Low bandwidth consumption with high transmission efficiency gave birth to the MQTT communication protocol. Moreover, the PDF reporting features performed ideally without the computer slowing down.

Excellent For real-world uses of environmental monitoring, the system revealed minimal lag, excellent performance, and scalable performance generally. Further fusing of artificial intelligence and Internet of Things elements increases its ability for smart decision-making and next-generation scalability.

Modular building lets the design scale rather well. The database level might manage expanding time-series data; Thanks to MQTT-based communication, several sensor nodes are easy to integrate. The system could be spread across several geographic locations without considerable performance reduction to keep track of more water quality parameters.

## 7. CONCLUSIONS

This paper presents an IoT-based water quality monitoring system that combines low-cost hardware with feature-rich software capabilities. The system utilizes a Raspberry Pi Zero, ADS1115 ADC, pH sensor, and temperature sensor to enable reliable real-time monitoring of water parameters. The collected data is securely stored and further analyzed to compute weekly averages and predict water quality trends using linear regression. These results are displayed through a multilingual graphical user interface, enhancing accessibility for diverse users.

Additional features such as automated PDF report generation and geospatial visualization of nearby water bodies further improve the system's usability. The overall design is portable, energy-efficient, and cost-effective compared to most commercial solutions that monitor only a single parameter.

Future enhancements may include the integration of additional sensors such as turbidity, total dissolved solids (TDS), and dissolved oxygen to provide more comprehensive water quality analysis. The system can also be extended using solar power for continuous operation in remote areas. Communication can be improved by adopting long-range technologies such as LoRa for wider coverage.

Further improvements may involve the use of advanced time-series machine learning models for more accurate predictions and the development of a mobile application for real-time monitoring and alerts. Integration with GPS modules and deployment on floating platforms can also enable large-scale environmental monitoring across lakes, rivers, and reservoirs.

## 8. ACKNOWLEDGEMENT

I would like to express my sincere and profound gratitude to all those who have contributed to the successful completion of this research paper. First and foremost, I extend my deepest appreciation to my respected guide and mentor for their continuous support, expert guidance, and valuable insights throughout the course of this study. Their constructive suggestions and encouragement have been instrumental in shaping the direction and quality of this work.

I am also grateful to my institution for providing the necessary infrastructure, academic resources, and a conducive environment for carrying out this research effectively. I would like to acknowledge the support of my colleagues and peers for their cooperation, thoughtful discussions, and constructive feedback, which have significantly enhanced the quality of this paper.

Furthermore, I express my heartfelt thanks to my family for their constant encouragement, patience, and understanding during the entire research process. Their unwavering support has been a source of motivation and strength.

Finally, I extend my thanks to all those who, directly or indirectly, contributed to the completion of this research work.

## 9. REFERENCES

- [1] H. Patel and D. Patel, "IoT-based water quality monitoring system," *Mater. Today: Proc.*, 2021, doi: 10.1016/j.matpr.2021.07.291.
- [2] "Real-time water quality monitoring system using IoT and artificial intelligence," *Int. J. Eng. Res. Technol.*, vol. 9, no. 6, 2020.
- [3] R. Suresh and R. Ranjith, "Environmental machine learning-based water quality monitoring system," *Environ. Technol. Innov.*, 2024, doi: 10.1016/j.eti.2024.103123.
- [4] D. Mistry and D. Shah, "Review on artificial intelligence-based water quality monitoring systems," *Int. J. Sci. Eng. Res.*, vol. 10, no. 4, pp. 1–8, Apr. 2019. [Online]. Available: <http://www.ijser.org/researchpaper/A-Review-on-AI-based-WaterQuality-Monitoring-System.pdf>
- [5] A. Gupta et al., "AquaSight: Automatic water impurity detection utilizing convolutional neural networks," *arXiv preprint arXiv:1907.07573*, Jul. 2019.
- [6] J. Wang, C. Zha, W. Wang, G. Chen, and L. Liu, "Simulation-based optimization of urban water storage tank operations: Balancing hydraulic stability, water quality, and energy conservation," *Journal of Water Process Engineering*, vol. 71, p. 107162, 2025.
- [7] P. Verma and P. Mehta, "Emerging Trends in Real-Time Water Quality Monitoring and Sanitation Systems," *Sustaining Water Quality-From Local Challenges to Global Solutions*, 2025.