INTERNATIONAL JOURNAL OF RESEARCH, EXPLORATION & KNOWLEDGE (IJREK)
Official Scholarly Journal Portal: **https://www.ijrek.org** | DOI: **https://doi.org/10.5281/zenodo.23482448**

SMART WATER QUALITY MONITORING SYSTEM

Mr. Suryvanshi A. M[1], prajakta gore[2]

[1] Faculty Guide & Supervisor | [2] Dattakala Group Of Institutions Faculty Of Engineering Swami - Chincholi tal. Daund dist. Pune, India

Corresponding author: **prajaktagore261900@gmail.com**

ARTICLE ID	ARTICLE TYPE	RESEARCH AREA	PUBLICATION
IJREK-2026-00100	Research Article	Computer Science	Volume 1, Issue Issue 1 (2026)
RECEIVED	**REVISED**	**ACCEPTED**	**PUBLISHED**
19 October 2026	10 November 2026	19 October 2026	11 November 2026

Abstract

Water quality monitoring is essential for public health, environmental protection, agriculture, and industrial operations. Conventional monitoring often depends on periodic sample collection and laboratory analysis, which can create delays between a change in water conditions and the response to it. This paper proposes a Smart Water Quality Monitoring System that combines water-quality sensors, a microcontroller, communication connectivity, a data-storage service, and a dashboard to support frequent measurements and timely alerts. The proposed prototype focuses on parameters such as pH, turbidity, and total dissolved solids (TDS), with optional temperature sensing. Readings are validated, time-stamped, compared with configured thresholds, and presented as trends. The system is intended as a screening and decision-support tool; it does not replace certified laboratory testing or establish potability on its own. The paper describes the architecture, workflow, implementation plan, evaluation metrics, potential applications, and limitations. A conceptual trend chart is included to demonstrate how sensor readings may be visualised; it is illustrative and is not presented as collected experimental data.

Keywords: *Research; Technology*

2. PROBLEM STATEMENT AND OBJECTIVES

Periodic sampling is important, but it may not reveal short-lived changes between sampling intervals. Manual logging can also be inconsistent, and results may not be available to a responsible person immediately. The project addresses these operational limitations by creating a system that records selected indicators at a configured interval and flags readings that fall outside project-defined limits. The main objectives are: To measure selected indicators such as pH, turbidity, and TDS using suitable sensors. To read sensor signals through a microcontroller and apply basic validation and calibration. To transmit timestamped measurements to a dashboard or database. To display current values and historical trends in a readable format. To generate alerts when readings cross configured thresholds or sensor faults are detected. To evaluate measurement consistency, data delivery, alert latency, and system availability.

3. LITERATURE REVIEW

Research and deployed monitoring systems have explored sensor networks, embedded devices, wireless communication, and cloud dashboards for environmental observation. Low-cost microcontrollers such as Arduino-compatible boards and ESP32 modules are widely used in educational IoT prototypes because they support sensor interfaces and network connectivity. Cloud platforms and lightweight web services can store readings and provide dashboards for remote access. A recurring engineering challenge is that low-cost sensors may drift, require frequent calibration, and respond differently to temperature or sample composition. Turbidity and TDS measurements are useful indicators but do not identify every contaminant. Similarly, pH alone cannot establish that water is safe to drink. Therefore, a practical system should communicate uncertainty, maintain calibration records, and distinguish an alert from a formal water-quality certification. The proposed design combines these established concepts into a modular workflow. Its contribution is the integration plan and evaluation framework for an academic prototype rather than a claim of a new sensor or validated public-health diagnostic method.

4. SCOPE AND ASSUMPTIONS

The prototype is scoped to monitoring selected indicators at a fixed site. Sensor choice, thresholds, sample interval, and communication method depend on the intended source and available hardware. Thresholds must be selected from applicable standards and project requirements; they should not be treated as universal for every use. Any deployment intended to guide drinking-water decisions must be supported by suitable calibration, validation, and laboratory confirmation.

5. PROPOSED SYSTEM ARCHITECTURE

The system is organised into five layers: sensing, embedded processing, communication, data services, and user interface. The modular arrangement allows a sensor or communication method to be replaced without redesigning the entire application. Water source River / tank / well Sensors pH, turbidity, TDS Controller ESP32 / Arduino Cloud / server Store & analyse Dashboard Live status Alerts SMS / app / email SMART WATER QUALITY MONITORING — HIGH-LEVEL ARCHITECTURE

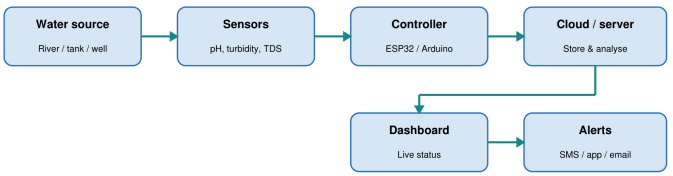


Figure 1. High-level architecture of the proposed monitoring system.

5.1 SENSING LAYER

A pH sensor measures acidity or alkalinity, a turbidity sensor estimates the cloudiness caused by suspended particles, and a TDS sensor estimates the concentration of dissolved ionic material through electrical conductivity-based measurement. A temperature sensor may be added to contextualise readings and support compensation where the sensor design permits it. Each sensor requires correct installation, conditioning, and calibration.

5.2 EMBEDDED PROCESSING LAYER

An ESP32 or equivalent microcontroller reads the sensor interface, applies simple range checks and filtering, attaches a timestamp, and prepares a measurement record. Invalid or missing readings should be marked as unavailable rather than silently converted into plausible values. The device should also record sensor status and communication failures where possible.

5.3 COMMUNICATION AND DATA SERVICES

Wi-Fi is suitable where network access is available; other links may be considered for remote locations. A web API or IoT platform receives readings and stores them with the device identifier, timestamp, unit, calibration metadata, and quality flag. Access controls and secure transport should be used when the deployment supports them.

5.4 USER INTERFACE

The dashboard presents current measurements, recent trends, last-update time, and alert history. Clear labels and units reduce interpretation errors. The interface should indicate when data is stale or when a sensor is disconnected.

6. SYSTEM WORKFLOW AND METHODOLOGY

1. Sample Collect water 2. Sense Read sensors 3. Validate Filter / calibrate 4. Compare Threshold rules 5. Notify Dashboard / alert DATA-TO-ALERT WORKFLOW



Figure 2. Processing sequence from sample measurement to alert.

6.1 MEASUREMENT CYCLE

At a configured interval, the controller obtains readings from each connected sensor. The software checks whether values are present, within the sensor's supported range, and stable enough for the chosen sampling procedure. A short moving average or median filter may reduce random fluctuations, but filtering parameters must be documented so that real changes are not hidden.

6.2 CALIBRATION AND VALIDATION

Calibration should use appropriate reference solutions or standards according to the sensor manufacturer's instructions. Calibration date, coefficients, and operator notes should be stored. Validation should compare prototype readings with a suitable reference instrument or laboratory result across multiple samples and conditions. A single calibration point may be insufficient for a sensor's full operating range.

6.3 THRESHOLD AND ALERT LOGIC

The application compares validated measurements with configurable limits chosen for the intended monitoring context. When a reading crosses a limit, the system creates an event containing the parameter, observed value, timestamp, device identifier, and configured threshold. Hysteresis or repeated-sample confirmation can reduce repeated alerts caused by noisy readings. Alerts indicate that review or confirmatory testing may be needed; they do not by themselves prove contamination.

6.4 DATA MODEL

A measurement record can contain: device_id, parameter, value, unit, timestamp, calibration_version, sensor_status, and quality_flag. An alert record can contain: alert_id, parameter, observed_value, threshold, event_time, acknowledgement_status, and resolution_note. Keeping measurement and alert records separate supports auditing and later analysis.

6.5 PROPOSED TECHNOLOGY STACK

Hardware: ESP32 or Arduino-compatible controller and compatible sensors. Firmware: C/C++ or a suitable embedded development framework. Backend: a lightweight REST API or IoT platform. Storage: a relational or time-series database. Frontend: a responsive web dashboard with charts and alert history. The final stack should be chosen based on budget, connectivity, maintainability, and team skills.

7. MONITORING DASHBOARD AND DATA VISUALISATION

The dashboard should help a user understand both the current state and how readings change over time. Each parameter should have its own unit, timestamp, and status indicator. The application should avoid presenting a single green/red label as a complete judgement of drinking-water safety. Sample index (illustrative) pH ILLUSTRATIVE SENSOR TREND — NOT MEASURED FIELD DATA Reference band (context-dependent)

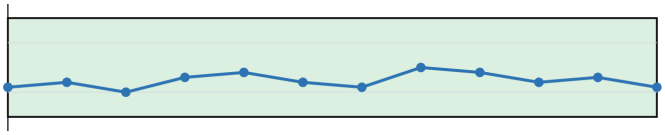


Figure 3. Conceptual pH trend for interface demonstration only. Values are illustrative, not experimental results or a water-safety

7.1 SUGGESTED DASHBOARD COMPONENTS

Current-value cards for pH, turbidity, TDS, and optional temperature. Time-series plots with selectable time windows. Status indicators for sensor connectivity and the age of the latest reading. An alert table with threshold, event time, acknowledgement, and notes. Export of timestamped readings for project evaluation and reporting.

7.2 DATA QUALITY INDICATORS

A reading should be shown as stale if no update arrives within the configured period. Sensor errors, calibration expiry, and missing values should be visually distinct from a genuine threshold breach. Where a reading is smoothed, the dashboard can retain the raw measurement for review and show the filtered value separately.

7.3 ACCESSIBILITY AND USABILITY

Colour should not be the only way to communicate status; text labels and icons should accompany it. The interface should be usable on a mobile screen and should show measurement units consistently. Alert messages should explain what happened and what follow-up action is recommended, such as checking the probe or arranging a confirmatory sample.

8. IMPLEMENTATION PLAN

Table 1. Example components and responsibilities

Component	Purpose	Key consideration
pH sensor and probe	Estimate acidity/alkalinity	Buffer calibration; probe care
Turbidity sensor	Estimate suspended-particle cloudiness	Optical fouling; sample conditions
TDS / conductivity sensor	Estimate dissolved ionic content	Temperature and calibration effects
ESP32 / controller	Read sensors and manage connectivity	Power stability; safe input ranges
API and database	Store and serve timestamped readings	Security, backups, retention
Web dashboard	Display trends and alerts	Usability; stale-data handling

8.1 RELIABILITY AND SECURITY

The firmware should recover from temporary network failures and avoid losing buffered readings where memory permits. Device credentials should not be hard-coded into public source repositories. The backend should validate incoming data, restrict administrative actions, and use encrypted transport where supported. Physical enclosures should protect electronics from moisture while allowing probes to contact the sample as intended.

9. EVALUATION PLAN

Evaluation should be performed using recorded tests rather than assumed performance figures. The following metrics provide a practical basis for assessing the prototype. Metric Method Purpose Measurement agreement Compare sensor readings with reference instrument/lab

RESULTS

TABLE II

Metric	Method	Purpose
Measurement agreement	Compare sensor readings with reference instrument/lab results	Estimate bias and error
Data delivery rate	Received records divided by expected records	Assess communication reliability
Alert latency	Time from threshold event to alert display	Assess timeliness
Availability	Operational time divided by test duration	Assess uptime
False-alert review	Inspect alerts against validated reference readings	Tune thresholds and filtering

11. LIMITATIONS AND RISK MANAGEMENT

Low-cost probes can drift, foul, or fail; connectivity and power may be intermittent; and measurements may be affected by temperature, suspended material, and sample handling. pH, turbidity, and TDS do not detect every pathogen, toxic chemical, or emerging contaminant. A dashboard alert is therefore an indication for investigation, not a guarantee of safety. The project should document calibration, cleaning, sensor replacement, data gaps, and reference testing. Any drinking-water decision should follow applicable local standards and competent laboratory advice.

12. EXPECTED OUTCOMES

The planned outcome is a working prototype that demonstrates sensor data acquisition, structured logging, a responsive dashboard, and threshold-based notification. Quantitative claims about accuracy, cost savings, or response time should only be made after the evaluation phase and should be supported by recorded measurements.

13. FUTURE SCOPE

Future improvements may include additional sensors selected for a specific use case, solar power for remote deployments, local buffering during network outages, secure multi-site management, role-based access, and predictive analysis for detecting unusual trends. More advanced analytics should be introduced only after sufficient high-quality reference data are collected. Integration with laboratory information systems or official reporting channels would require appropriate agreements, validation, and security review. 14. Conclusion This paper presents a proposed Smart Water Quality Monitoring System using sensors, an embedded controller, connectivity, data storage, and a dashboard. The architecture supports frequent observation of selected water-quality indicators and can help users notice unusual changes more quickly than periodic manual logging alone. The workflow includes sensor validation, calibration records, threshold-based alerts, and visualisation of historical data. The design remains a monitoring aid rather than a substitute for certified laboratory analysis. A successful implementation should be judged through reference comparisons, data-delivery tests, alert-latency measurements, and transparent reporting of limitations.

REFERENCES

[1] World Health Organization, Guidelines for Drinking-water Quality: Fourth Edition Incorporating the First and Second Addenda. Geneva: WHO, 2022.

[2] Bureau of Indian Standards, IS 10500: Drinking Water — Specification, second revision and applicable amendments. New Delhi: BIS. Consult the current official edition for requirements.

[3] Espressif Systems, ESP32 Series Datasheet. Available from the official Espressif documentation portal.

[4] Arduino, Arduino Documentation. Available from the official Arduino documentation website.

[5] World Health Organization, Water Quality and Health: Review of Turbidity. WHO technical resources on drinking-water quality and operational monitoring.

[6] U.S. Environmental Protection Agency, Water Quality Monitoring. EPA resources describing monitoring approaches and water-quality indicators.

[7] International Organization for Standardization, ISO 7027-1:2016, Water Quality — Determination of Turbidity. Geneva: ISO, 2016.

[8] Analog Devices, Technical Resources on Conductivity and Sensor Signal Conditioning. Consult device-specific documentation for implementation details.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on **https://www.ijrek.org** | Article ID: IJREK-2026-00100