TITLE PAGE

Project Title: AquaGuard - IoT Based Real Time Water Quality Monitoring and Alerting System

Author: [Student Name]

Department: [University / Course Name]

Institution: [University Name]

Date: [Month, Year]

1. INTRODUCTION

Access to clean water is a critical public health requirement. Manual testing of water quality is infrequent, slow, and often reactive. AquaGuard provides a continuous water quality monitoring solution using IoT sensor nodes capable of measuring parameters such as pH, turbidity, temperature, and Total Dissolved Solids. The system publishes sensor data over MQTT to a backend platform for real time visualization, threshold based alerting, and long term analysis.

2. SYSTEM ARCHITECTURE OVERVIEW

AquaGuard consists of distributed sensing units deployed across water sources and storage systems. Sensor data is transmitted securely to a cloud or local processing layer, stored in a time series database, and visualized through dashboards with automated alerts.

3. HARDWARE DESIGN AND COMPONENTS

3.1 Component List

Microcontroller: ESP32 with WiFi connectivity Sensors: pH sensor probe with analog interface Turbidity sensor (optical reflection based) TDS sensor for dissolved solids Waterproof temperature sensor (DS18B20) Supporting hardware: Voltage regulator module Waterproof casing (IP67 recommended) Power supply via DC adapter or solar unit

3.2 Circuit Integration

Step 1: Interface pH amplifier output to ESP32 ADC pin. Step 2: Connect turbidity sensor signal to analog input. Step 3: Connect DS18B20 to digital pin using one wire protocol. Step 4: Provide stabilized 5V input using regulator. Step 5: Waterproof all cable entry points to prevent corrosion.

3.3 Hardware Reliability Testing

Validation steps include: Sensor calibration using buffer solutions for pH Optical clarity testing for turbidity sensor Submersion tests for casing integrity Long duration power stability test

4. SOFTWARE DEVELOPMENT

4.1 Firmware Workflow

ESP32 firmware responsibilities: Initialize WiFi and handle reconnection Perform sensor sampling at fixed intervals Apply calibration coefficients to raw readings Package readings into JSON Publish data to MQTT broker Implement watchdog and failure recovery

4.2 Data Schema and Topic Structure

MQTT topics follow structured naming: aquaguard/site1/ph aquaguard/site1/turbidity Payload format: { sensor: ph, value: 7.12, unit: pH, timestamp: 1700000000 }

4.3 Backend Processing Microservice

A Python based service performs: Subscription to all AquaGuard topics Validation and range sanity checks Transformation into InfluxDB line protocol Error logging and retry handling

5. PLATFORM AND DATABASE SETUP

5.1 MQTT Broker Configuration

Mosquitto broker settings: Enable authentication and access control lists Use persistent sessions for reliability Optional TLS certificates for secure communication

5.2 InfluxDB Setup

Steps include: Create bucket named aquaguard Generate write token Define retention rules based on storage duration Enable continuous queries for rollups

6. DASHBOARDING AND ALERTING

6.1 Dashboard Views

Grafana dashboards include: Real time parameter charts for pH and turbidity Daily average TDS and temperature profiles Multi site comparison panels Historical analysis for trend detection

6.2 Alert Rules

Alert conditions configured: pH outside safe range of 6.5 to 8.5 Turbidity exceeds 5 NTU for more than 5 minutes Sudden TDS spike indicating contamination Alerts delivered via SMS, email, or webhook

7. IMPLEMENTATION PLAN

7.1 Field Installation

Steps: Mount sensors in flowing water for accurate sampling Ensure probes are fully submerged Avoid placement near sediment disturbance zones Secure node using mounting bracket

7.2 Firmware Deployment

Steps: Flash firmware via USB Load WiFi and site credentials Test MQTT publish using test topics Enable remote configuration support

7.3 Backend Deployment

Steps: Deploy microservice as systemd or Docker container Verify end to end data flow Enable logging rotation and monitoring

8. TESTING AND QUALITY VALIDATION

8.1 Functional Testing

Tests include: Calibration accuracy verification MQTT reliability under network drop Dashboard refresh latency checks

8.2 Environmental Stress Testing

Simulate: Temperature fluctuations Prolonged submersion exposure Power cycling and brownout conditions

8.3 Performance Metrics

Measured parameters: End to end latency Data packet loss rate Microcontroller resource usage

9. DEPLOYMENT STRATEGY

9.1 Local Deployment

Run full stack on Raspberry Pi for small facilities. Includes MQTT, InfluxDB, Grafana, and microservice.

9.2 Cloud Deployment Option

Use cloud VM for geographically distributed sites: TLS enforced MQTT Remote access dashboards Multi site scalability

10. MAINTENANCE AND FUTURE ENHANCEMENTS

Future improvements: Solar powered autonomous units Machine learning based contamination prediction Mobile application for onsite inspections Automated probe cleaning mechanism

11. CONCLUSION

AquaGuard demonstrates a realistic and deployable IoT based water quality monitoring solution with strong implementation detail across hardware, firmware, data pipelines, and visualization.

12. REFERENCES

[1] WHO Water Quality Guidelines. [2] Mosquitto MQTT Documentation. [3] InfluxDB and Grafana References.