**CHAPTER 1 INTRODUCTION**

**1.1 INTRODUCTION**

The Aquaculture Automation and Monitoring System is an innovative project aimed at transforming the aquaculture industry by addressing key challenges related to resource efficiency, water quality management, and fish health. Traditional aquaculture practices often suffer from inefficiencies such as overfeeding, inconsistent monitoring, and reliance on labor-intensive processes. These issues result in increased operational costs, environmental degradation, and sub-optimal fish growth. This project leverages cutting edge technologies, including the Internet of Things (IoT) and to provide a comprehensive and intelligent solution to these challenges.

At its core, the system combines IoT-based real-time monitoring to ensure precise control and automation in aquaculture environments. By utilizing predefined thresholds, the system can identify anomalies in water quality and automatically trigger corrective actions. This intelligent system reduces dependency on human intervention, ensures resource efficiency, and provides a stable aquatic environment for optimal fish health and growth.

The Aquaculture Automation and Monitoring System is designed as a multi-functional and scalable platform that integrates seamlessly into aquaculture operations. It offers a robust set of features tailored to meet the diverse needs of fish farmers:

1. **Automated Water Quality Monitoring**: IoT sensors continuously measure critical parameters such as pH, temperature, TDS, and dissolved oxygen. Alerts are generated when values cross predefined thresholds, enabling proactive interventions to maintain optimal water conditions.
2. **Smart Feeding System**: The integration of camera allows the system to analyze fish behavior and automate feeding schedules. This reduces overfeeding, minimizes waste, and ensures balanced nutrition for the fish.
3. **Resource Optimization**: Actuators for oxygenation, filtration, and water purification are triggered automatically based on real-time data, ensuring efficient use of resources while reducing operational costs.
4. **User-Friendly Mobile Application**: A dedicated app enables farmers to monitor water quality, control automated systems, and receive alerts from anywhere in the world, providing convenience and accessibility.
5. **Scalability and Sustainability**: Designed to accommodate farms of varying sizes, the system promotes sustainable aquaculture practices by minimizing environmental impact and ensuring long-term resource conservation.
6. **Ease of Use**: The system is built to be user-friendly, allowing non-technical users to operate and manage their aquaculture operations with minimal training.

By addressing the critical inefficiencies of traditional aquaculture, this system sets a new benchmark in the industry. Its ability to automate processes, adapt to environmental changes, and provide actionable insights ensures that fish farming becomes more productive, sustainable, and cost-effective. Whether it’s maintaining optimal water quality, improving feeding efficiency, or providing real-time data access, this project redefines the aquaculture experience, empowering farmers to achieve better outcomes with less effort.

**1.1.1 OBJECTIVE**

The objective of this project is to design and implement a smart aquaculture monitoring and control system using IoT technologies. The system continuously monitors critical water parameters such as pH, temperature, TDS (Total Dissolved Solids) and water level using various sensors. It also observes fish behavior through a camera module. The collected data is transmitted in real-time to a cloud platform for storage and analysis. Based on the analyzed data, the system can automatically control the feeding mechanism, oxygen supply, and water flow (outlet) using actuators like servo motors, relays, and pumps. This intelligent and automated approach aims to reduce manual labor, optimize resource usage, improve fish health, and increase productivity in a cost-effective and sustainable manner.

# **1.2 SCOPE**

The scope of this IoT-based aquaculture project encompasses the design and implementation of a comprehensive system for optimizing fish farming practices through advanced monitoring and automation technologies. The project aims to integrate IoT devices for real-time data collection on water quality and fish behavior, facilitating data-driven decision-making to improve feeding strategies and enhance fish health. Additionally, the project includes the development of a user-friendly app for visualizing key metrics and alerts, enabling farm operators to monitor their aquaculture systems effectively. Overall, this initiative seeks to promote sustainable aquaculture practices, enhance productivity, and support the long-term viability of fish farming operations.

# **CHAPTER 2 PROBLEM DEFINITION**

Aquaculture faces significant challenges in achieving optimal fish growth due to inefficient feeding strategies and inadequate water quality management, with overfeeding escalating operational costs and compromising fish health. The reliance on manual monitoring of essential parameters is labor-intensive and often unreliable, hindering fish farm operators' ability to make informed, data-driven decisions. These inefficiencies undermine productivity and sustainability in contemporary fish farming. To address these challenges, integrating IoT-cloud based technologies and android app provides a promising solution by enabling real-time data collection, optimizing feeding strategies, and effectively monitoring water quality, thereby enhancing productivity and promoting healthier aquatic environments.

# **CHAPTER 3 LITERATURE REVIEW**

### **PROTOTYPE MODEL OF POLY HOUSE FARMING USING SENSOR AND IoT TECHNOLOGIES**

In recent years yield from the agriculture fields was affected by the climate changes due to pollution and global warming etc. Another reason is that still our farmers are practicing conventional practices. On the other hand resources like land and water are diminishing These are the main challenges concentrated in this research by using green technologies i.e. poly house farming. Hence the work in this paper was focused on improving the yield from the agricultural domain by implementing end-end solution to control parameters such as temperature, humidity, soil moisture, CO2, light intensity, rainfall, tank level etc. This work is also concentrated on rain water harvesting. The data from the sensors was analyzed using cloud server and subscriptions are sent to the end users using GSM and Android app. It is observed that yield was increased with high quality and optimal resource utilization.

### **A Novel Methodology for Monitoring and Controlling of Water Quality in Aquaculture using Internet of Things (IoT)**

We have designed Internet of Thing (IoT) system based on for monitoring and controlling the water parameters in the aquaculture. The system can detect and control the parameters such as temperature, pH value, dissolved oxygen, water level, foul smell detector and ammonia in the water. The sensors nodes gather the real time data from the water and send it to the aurdino processor for processing. If the measured parameters exceeds the desired range then the processor activates the corresponding controller for taking necessary action. The measured values from the sensors are also sent to the cloud using wi-fi modem and can be viewed in the control room. Also the values are sent as short messages to the concern person using GSM modem. The proposed system is compatible and can be used for any type of aquaculture system.

### **Knowledge Based Real Time Monitoring System for Aquaculture Using IoT**

Internet of things is one of the rapidly growing fields for delivering social and economic benefits for emerging and developing economy. The field of IOT is expanding its wings in all the domains like medical, industrial, transportation, education, mining etc. Now-adays with the advancement in integrated on chip computers like Arduino, Raspberry pi the technology is reaching the ground level with its application in agriculture and aquaculture. Water quality is a critical factor while culturing aquatic organisms. It mainly depends on several parameters like dissolved oxygen, ammonia, pH, temperature, salt, nitrates, carbonates etc. The quality of water is monitored continuously with the help of sensors to ensure growth and survival of aquatic life. The sensed data is transferred to the aqua farmer mobile through cloud. As a result preventive measures can be taken in time to minimize the losses and increase the productivity.

### **An application of internet of things on sustainable aquaculture system**

Aquaculture is one of the human important food sources. This paper proposed the prototype of the framework, Sustainable Fish-Farming System(SFFS), which can make the aqua-farming system more sustainable via applying the Internet of things(IOT) to reduce the need of energy for controlling the environment. The SFFS integrates the solar-farm and fish-farm to reduce the extra energy input. In addition, the lighting of LEDs is used to support the photosynthesis in the night-time. This way is more energyefficient than the traditional pumping. Furthermore, this prototype shows the qualitative availability of SFFS.

### **IoT Based Automated Fish Farm Aquaculture Monitoring System**

Internet of Things (IoT) is a very fast growing technology and the field of IoT is extending its wings in every one of the areas today. With the progression in computers like Arduino, Raspberry pi, the innovation is achieving the ground level with its application in farming and aquaculture. In this work, we have outlined and actualized monitoring of water quality of aquaculture utilizing Raspberry Pi, Arduino, various Sensors, Smartphone Camera and Android application. Water quality parameters used in this work are Temperature, pH, Electrical Conductivity and Colour. Sensor acquisition is conducted by Arduino and Raspberry Pi is used as data processing device as well as server. Photo acquisition is also performed by Raspberry Pi with the help of the smartphone camera to detect the colour of the water. Android phone is used as the terminal device. A user can monitor the water condition using an android app through Wi-Fi within Wi-Fi range and through Internet from anywhere in the world. Some analysis is performed with the four parameters value to determine the overall approximate condition of the water and required action. Every feature in this checking gadget can work legitimately and easily.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. No** | **AUTHOR** | **YEAR** | **TECHNIQUE** | **ADVANTAGES** |
| 1. | SAJAL SAHA, RAKI BUL HASAN RAJIB, SUMAIYA KABIR | 2018 | It uses Raspberry pi, Arduino, various sensors, android application and smartphone camera for water quality monitoring. | It finds a way to give a better result for low cost than any other available systems. |
| 2. | Ria Gemel B. Palconit, Ronnie S. Concepcion II, Rogelio Ruzcko Tobias, Jonnel Alejandrino, Vincent Jan D. Almero, Argel A. Bandala, Ryan Rhay P. Vicerra, Edwin Sybingco, and Elmer P. Dadios | IEEE  2023 | IoT, microcontroller,google sheet,matlab,sensors network. | The system effectively monitored water quality parameters like pH, dissolved oxygen, temperature, and TDS with accurate data transmission and minimal video recording errors, demonstrating its feasibility for remote fish tank monitoring. |
| 3. | K. Raghu Sita Rama Raju, G. Harish Kumar Varma / Knowledge-Based Real-Time Monitoring System for Aquaculture Using IoT | 2017 | IoT-based system with Raspberry Pi 3 Solar-powered with battery backup.Sensors for dissolved oxygen, ammonia, pH, nitrates, carbonates, bicarbonates, and temperature.Data sent to cloud and alerts sent to farmer’s app. | Replacing Raspberry Pi with  ESP8266-12E could reduce system cost while maintaining functionality.ESP8266 is more power-efficient for transmitting sensor data, ideal for solarpowered setups.Cloud integration can still be achieved with ESP8266, making it a lightweight yet scalable solution. Recommended for projects requiring lower processing power robust communication. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4. | Aishwarya Girish Menon, Prabhakar M.I ioT-based Automated Pond Water Quality Monitoring System for  Aquaculture Farms | 2021 | Sensors for water quality: pH,Temperature, Electrical Conductivity,Ammonium Ion, Redox Potential,Dissolved Oxygen, Nitrite Ion.  Automated movement using DC motors and preprogrammed GPS coordinates. | GSM communication ensures usability in remote areas lacking internet connectivity. Real-time data collection and automated alerts improve efficiency and reduce risks associated with manual monitoring. |
| 5. | Ria Gemel B. Palconit, Ronnie S.  Concepcion II, Rogelio Ruzcko  Tobias, Jonnel Alejandrino, Vincent Jan D. Almero, Argel A. Bandala,Ryan Rhay P. Vicerra, Edwin Sybingco, and Elmer P. Dadios.Development of IoT-based Fish Tank Monitoring System | 2023 | The system effectively monitored water quality parameters like pH,dissolved oxygen, temperature, and TDS with accurate data transmission and minimal video recording errors, demonstrating its feasibility for remote fish tank monitoring. | The paper shows that the system uses pH, dissolved oxygen,temperature, and TDS sensors to accurately monitor water quality in real-time, enabling efficient and  automated fish tank management.. |
| 6. | Md. Ahsan Arif,Md.Rokonuzzaman Reza Angshu Bikash Mandal,Md. Appel Mahmud, Akib Farhana Mimi Shuma,Shah Syed Md. Fehir  Towards Developing an IoT-Based  Aquaculture Monitoring System | 2023 | IoT-based system with GSM SIM900A, Arduino temperature sensor, Analog pH Sensor, Turbidity Sensor ,SMS alerts, powered by 4.2V LiPo battery. | GSM communication ensures usability in remote areas lacking internet connectivity.  Real-time data collection and automated alerts improve efficiency and reduce risks associated with manual monitoring. |
| 7 | Smart IoT-based Feeder System for  Koi Fish (Cyprinus rubrofuscus) | 2021 | Uses ESP32 microcontroller for control.Monitors water quality with sensors. Servo motors automate feeding. Integrates with ThingSpeak for cloudbased monitoring. Mobile app provides real-time data and schedules.  Alerts for water quality deviations. | The ThingSpeak channel displays water temperature and level data, proving the system's effectiveness in monitoring conditions. Testing shows the system manages feeding and water conditions automatically for koi fish. The housing is made from cardboard for simplicity and efficiency. |
| 8 | BENAHMED TARIQ, BOUROUIS AMINA, and BENAHMED KHELIFA are affiliated with the  Department of Mathematics and | 2024 | The system uses sensors to monitor turbidity, water level, dissolved oxygen, salinity, ammonia, and carbon monoxide. It manages feeding, oxygenation, and water pumping with actuators, while Arduino and  Raspberry Pi handle data processing | The system automates monitoring of key water quality parameters, reducing manual labor and ensuring optimal conditions for fish health and productivity. Real-  time data collection and notifications via a mobile/web app |
|  | Computer Science, Faculty of Exact Sciences, Tahri Mohammed University of Bechar, Bechar, Algeria. |  | and IoT monitoring. | enable remote monitoring and quick interventions. The costeffective, low-power IoT design makes it accessible, while Firebase cloud storage ensures secure, scalable data management for informed decision-making. |
| 9 | IoT Based Automated Fish Farm Aquaculture Monitoring System[Text Wrapping Break]Sajal Saha, Rakibul Hasan Rajib, and Sumaiya Kabir are affiliated with the Department of Computer Science and Engineering. Sajal Saha and Rakibul Hasan Rajib are from Patuakhali Science and Technology University, Patuakhali, Bangladesh, while Sumaiya Kabir is from the Department of Computer Science and Engineering at Green University of Bangladesh, Dhaka. | 2018 | The system uses Arduino UNO for sensor data collection (pH, EC, temperature) and Raspberry Pi 3 for data processing and storage in a MySQL database. An Android app allows real-time monitoring, while Dataplicity provides remote access.The system uses Python for communication and data handling, and a web interface built with PHP for user interaction. | Advantages:[Text Wrapping Break]The system enables real time monitoring and automated alerts for water quality management, with a mobile app for easy control. It offers remote access and data logging for better decision-making, is affordable, and supports scalability for various aquaculture operations. The image-capturing feature adds a visual layer for water quality assessment. |

Table 3.1 Literature Review**CHAPTER 4 PROJECT DESCRIPTION**

The proposed **IoT-Based Aquaculture Monitoring System** aims to address the growing challenges in fish farming caused by inefficient water quality management and resource utilization. By integrating Internet of Things (IoT) technology and automation, the system continuously monitors critical aquatic parameters such as pH, dissolved oxygen, temperature, TDS, and water level using sensors. These parameters are vital for maintaining a healthy environment for aquatic life, as deviations can lead to poor fish health and reduced yields.

The system employs an ESP8266 microcontroller as the central processing unit to collect sensor data and control actuators like oxygen boosters, and inlet outlet motors. It automates processes such as oxygenation and water purification, ensuring optimal water quality. Additionally, an feeding system minimizes overfeeding, preventing water pollution and improving resource efficiency. Alerts and real-time data are sent to a cloud server, accessible to farmers via a mobile app, enabling remote monitoring and control. This system reduces manual labor, enhances fish health, and promotes sustainable aquaculture practices by improving operational efficiency and reducing environmental impact.

**4.1. SYSTEM DESIGN**

**FLOW CHART AND PROPOSED DESIGN FLOW CHART**

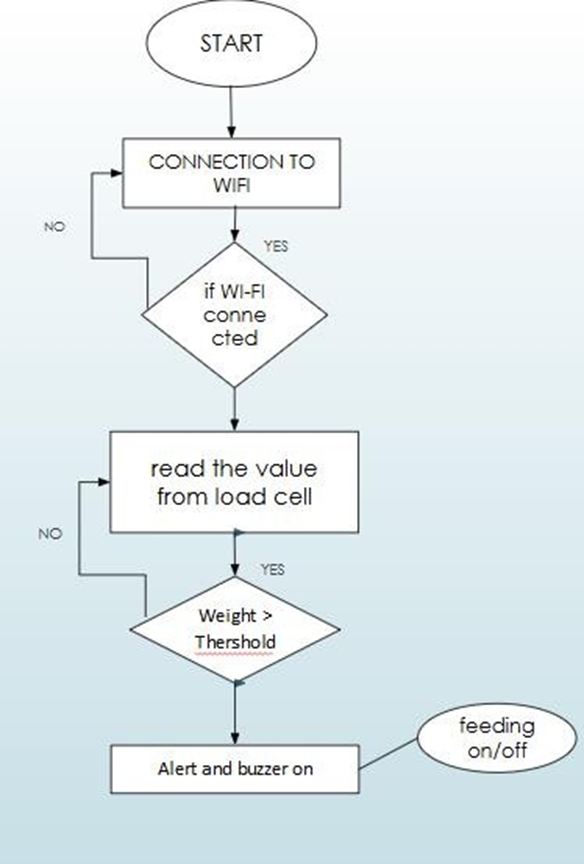


Fig 4.1 flow chart

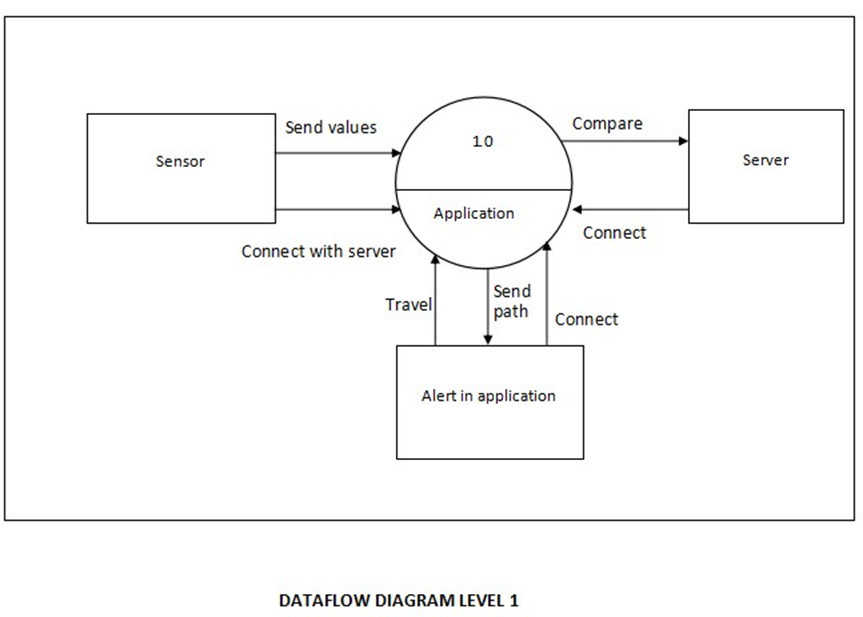
**DESIGN:** 

Fig 4.2 Data Diagram

**Data Flow and Communication**

1. **Sensor Data Collection**: Sensors capture real-time data (pH, temperature, water level, TDS). Data is transmitted to the ESP8266 for initial processing.
2. **Microcontroller Processing**: ESP8266 processes the data and sends it to the cloud via Wi-Fi.
3. **Cloud Integration**: Blink Cloud stores and processes the data, making it accessible for real-time monitoring and control.
4. **User Dashboard Communication**: The dashboard retrieves processed data from the cloud for visualization. Users send control commands (e.g., for feeding, oxygenation, or water pumping) via the dashboard.
5. **System Control Feedback**: ESP8266 executes control commands and updates the dashboard with the latest system status.

**4.2. ASSUMPTIONS AND DEPENDENCIES**

**4.2.1 Assumptions**

It is assumed that the aquatic environment remains within the operational range of the sensors used, such as temperature, pH, and turbidity sensors, to ensure accurate data collection. The ESP8266 microcontroller is expected to maintain a stable Wi-Fi connection to enable real-time data transmission to the cloud. A consistent and reliable power supply, whether from a battery or main source, is assumed to be available to support uninterrupted monitoring and automation processes. It is also presumed that the mobile application accurately receives and displays data from the cloud database, allowing users to monitor system parameters effectively. Furthermore, the user is expected to have access to a compatible smartphone or device capable of interacting with the mobile application or cloud dashboard. Lastly, it is assumed that environmental conditions, such as the presence of excessive algae or debris, do not significantly affect the performance and accuracy of the sensors.

**4.2.2 Dependencies**

**4.2.2.1 Software Dependencies**

* Arduino IDE for programming the ESP8266.
* Blynk for cloud-based data logging and visualization.
* Mobile application framework (Blynk App ).
* Sensor libraries and communication protocols (I2C).

**4.2.2.2 Hardware Dependencies**

* ESP8266 microcontroller (e.g., NodeMCU).
* Sensors: pH sensor, temperature sensor (DS18B20), , water level sensor.
* Actuators: Servo motor for automated fish feeding.
* Wi-Fi module (inbuilt in ESP8266) with internet access.
* Power supply unit (battery pack or USB adapter).
* Relay module (if controlling additional devices like aerators).

# **CHAPTER 5 REQUIREMENTS**

### **5.1 Functional Requirements**

**5.1.1 Real-Time Monitoring**

1. **Key Parameters:** The system will use sensors to track pH, temperature, TDS, dissolved oxygen, and fish behavior. These parameters will be continuously monitored, with real-time alerts sent if they deviate from set thresholds.
2. **Alerts:** You could implement this using cloud services like Blink IoT Core which can trigger notifications (e.g., app notification) when thresholds are crossed.

**5.1.2 Automated Feeding System**

1. **Feed Dispensing:** Based on water quality and fish behavior, automated feeding will be triggered. A microcontroller (ESP8266) will control a servo motor to dispense feed.

**5.1.3 Data Collection and Storage:**

1. Data from IoT sensors will be collected and stored in a cloud database Blink cloud for future analysis.

**5.1.4 User Interface:**

1. **Dashboard:** This app will visualize data, feeding schedules, alerts, and trends.

### **5.2 Non-Functional Requirements**

**5.2.1 Performance:** The system performs efficiently by processing sensor data quickly and updating it in real-time. The ESP8266 handles operations smoothly, while Blynk Cloud ensures fast and stable communication. Lightweight code and optimized data handling improve overall system responsiveness.

**5.2.2 Low Latency:** Real-time data should be updated with minimal delay (target < 5 seconds). Efficient coding, caching mechanisms, and optimized API usage can help.

1. **Concurrent Users:** Use scalable cloud solutions to ensure performance under increased user load.

**5.2.3 Scalability:**

1. **Sensor Addition:** The system should allow for easy addition of new sensors, possibly using a modular IoT architecture.
2. **Cloud Scalability:** Cloud platforms Blink should handle growing data and user volume without performance degradation.

**5.2.4 Reliability:**

1. **High Availability:** Implement fail-over mechanisms and use services with SLAs guaranteeing uptime (e.g., Blink 99.99% uptime guarantee).
2. **Backup & Recovery:** Data backup solutions (e.g., Blink Backup) and recovery mechanisms should be in place to prevent data loss.

**5.2.5 Usability:**

1. **User-Friendly UI:** The UI should be intuitive and require minimal training. It should also support mobile access to monitor the system on the go.
2. **Responsiveness:** Ensure that the app is responsive and accessible on different device types.

**5.3. HARDWARE AND SOFTWARE REQUIREMENTS**

### **5.3.1 Software:**

### **5.3.1.1 Arduino IDE:** The Arduino Integrated Development Environment (IDE) is an open-source platform used for writing, compiling, and uploading code to Arduino boards. It supports C/C++ programming and provides a simple interface for beginners to interact with micro controllers.

**5.3.1.2 Arduino-Based C++ Programming:** This refers to programming Arduino boards using a simplified version of C++ within the Arduino IDE. It includes predefined functions and libraries that make it easier to interact with hardware components such as sensors, motors, and LEDs for embedded system projects.

### **5.3.2 Hardware:**

**5.3.2.1 Micro controller (ESP8266):** The ESP8266 is a low-cost Wi-Fi micro controller used for controlling and monitoring devices remotely. It connects various sensors and components to the internet for real-time data transmission and automation in aquaculture systems.

**5.3.2.2 pH Sensor:** This sensor measures the pH level of water, helping monitor its acidity or alkalinity. Maintaining optimal pH levels is essential for fish health and water quality management.

**5.3.2.3 TDS Sensor:** The Total Dissolved Solids (TDS) sensor measures the concentration of dissolved substances in water, indicating its purity. High TDS levels may be harmful to aquatic life and require intervention.

**5.3.2.4 Water Temperature Sensor:** This sensor monitors the water temperature to ensure it stays within the suitable range for fish growth and survival. Fluctuations in temperature can stress fish and affect metabolism.

**5.3.2.5 Water Level Sensor:** Used to detect changes in the water level of the fish tank or pond. It helps prevent overflow or under fill conditions and ensures optimal water volume is maintained.

**5.3.2.6 Relay:** A relay acts as a switch that allows the microcontroller to control high-voltage devices such as motors or pumps. It enables automation by turning devices on or off based on sensor inputs.

**5.3.2.7 Servo Motor:** Servo motors are used for precise control of angular position. In aquaculture, they are often used to control feeding mechanisms or open/close valves.

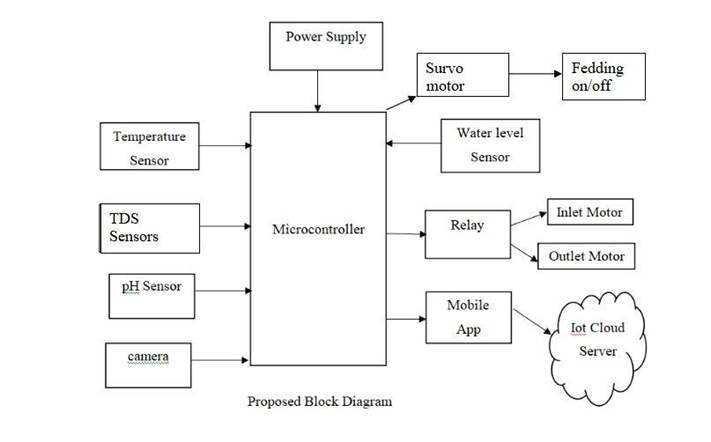
**5.3.2.8 Camera (ESP32-CAM):**  
The ESP32-CAM module includes a built-in camera that captures images or videos for monitoring fish behavior and tank conditions. It can be triggered automatically or manually for visual inspection.

**5.3.2.9 Oxygen Motor (Aerator):** This motor is used to oxygenate the water, ensuring an adequate supply of dissolved oxygen for fish respiration and preventing oxygen depletion.

**5.3.2.10 Outlet Motors (Pump-in and Pump-out):** These motors are responsible for water circulation pumping fresh water into the tank and removing waste water. They help maintain clean water conditions and control pollution levels.

# **CHAPTER 6 Methodology**

# 



**Fig 6.1** Propossed Block Diagram

The Microcontroller (ESP8266) serves as the central unit in this aquaculture system, managing all sensor operations and ensuring seamless communication and control. The system utilizes Blynk Cloud and a third-party Blynk IoT application for real-time data transmission, enabling users to monitor and manage the system through a mobile interface.

The TDS Sensor measures the turbidity of the water by detecting suspended particles using light scattering. When TDS levels exceed the defined threshold, the motor driver activates the inlet and outlet motors to improve water quality. The motors stop automatically once the TDS levels return to acceptable limits. Simultaneously, the water level sensor ensures the correct water level is maintained during motor operation by continuously monitoring the pond's water status.

The pH Sensor is responsible for keeping the pH levels within the ideal range of 0 to 1. If the pH levels deviate, 0 for normal water and 1 for salty or impure water.send alert signal if the value is 1.

The Temperature Sensor monitors the water temperature continuously. If the temperature crosses the threshold, the system adjusts the water conditions to stabilize the temperature and maintain a suitable environment for the fish.

A camera is integrated into the system for fish behavior surveillance. The camera helps monitor signs of hunger or illness in the fish. Based on these observations, the feeding system can either be operated manually by the user via the mobile app or triggered automatically when hunger is detected. A servo motor dispenses precise amounts of food, reducing overfeeding and maintaining water quality.

Additionally, the camera assists in overseeing fish health to detect potential illnesses and alert users for timely action.

The system also includes automated water purification using the inlet and outlet motors whenever contamination is detected. Moreover, an oxygenation mechanism is activated to stabilize oxygen levels and maintain water quality if thresholds are exceeded. By integrating real-time monitoring, automated controls, and user interaction through the Blynk IoT app, the system provides an efficient, sustainable, and user-friendly approach to managing aquaculture environments.

**CHAPTER 7 EXPERIMENTATION**

The experimentation phase of the aquaculture project follows a structured process to ensure the efficient monitoring and management of water quality, fish behavior, and automated feeding. This chapter outlines the software and hardware implementation, integration, and observed challenges.

**7.1 SOFTWARE DEVELOPMENT**

**Objective**

To develop an IoT-based real-time monitoring system that collects, processes, and responds to water quality parameters and fish behavior using sensor data and mobile application interfaces.

**Sensor Integration and Data Processing**

**Steps:**

* Interfaced sensors (pH, TDS, temperature, water level) with the ESP8266 microcontroller using Arduino IDE (C++).
* Configured sensor calibration for accurate threshold detection.
* Collected real-time data from each sensor and transmitted it to the Blynk Cloud platform.

**Cloud Communication and Mobile App Integration**

**Steps:**

* Integrated the Blynk IoT application for real-time visualization and control.
* Designed mobile UI for live sensor data, alerts, motor controls, and camera access.
* Configured triggers and notifications for parameter breaches (e.g., abnormal pH or TDS levels).

**Automation Logic Implementation**

**Steps:**

* Programmed conditional logic for:
* Triggering inlet/outlet motors when TDS exceeds a set threshold.
* Activating oxygen motor when dissolved oxygen level drops.
* Sending alerts when water level is too high/low or pH is impure.
* Implemented servo motor-based automatic feeding system based on fish behavior detected via camera or user command.

**Camera and Feeding Control**

**Steps:**

* Integrated ESP32-CAM for fish surveillance.
* Enabled image capture when water quality deteriorates or feeding is required.
* Used servo motor to dispense controlled feed quantities upon manual or automated triggers.

**Testing and Optimization**

**Steps:**

* Tested real-time data transmission delays and system response across various network conditions.
* Fine-tuned threshold values and control delays for optimal environmental response.
* Conducted edge-case testing for power failure recovery, sensor disconnect, and unexpected behavior.

**Expected Results**

* Real-time monitoring and control through mobile app.
* Accurate automated operation of feeding, purification, and oxygenation systems.
* Alerts and logs maintained for all critical actions and thresholds.

**7.2 HARDWARE IMPLEMENTATION**

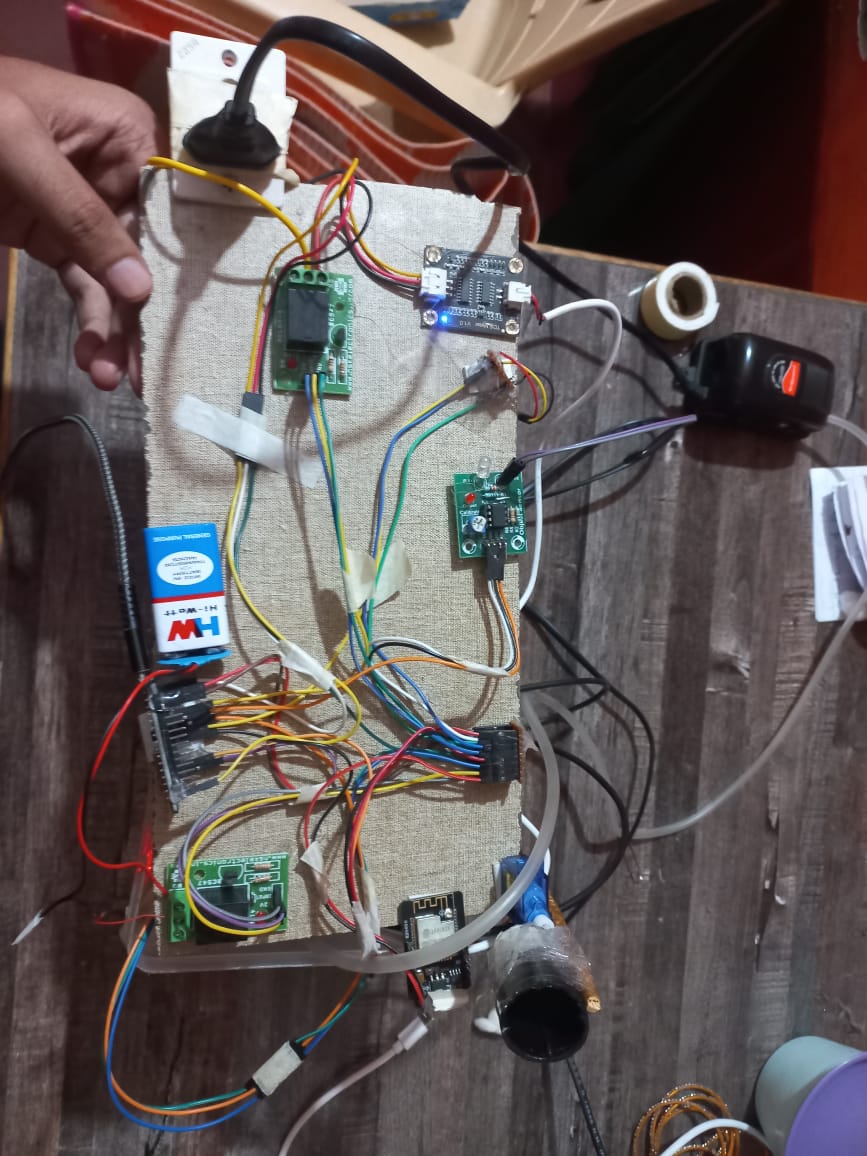
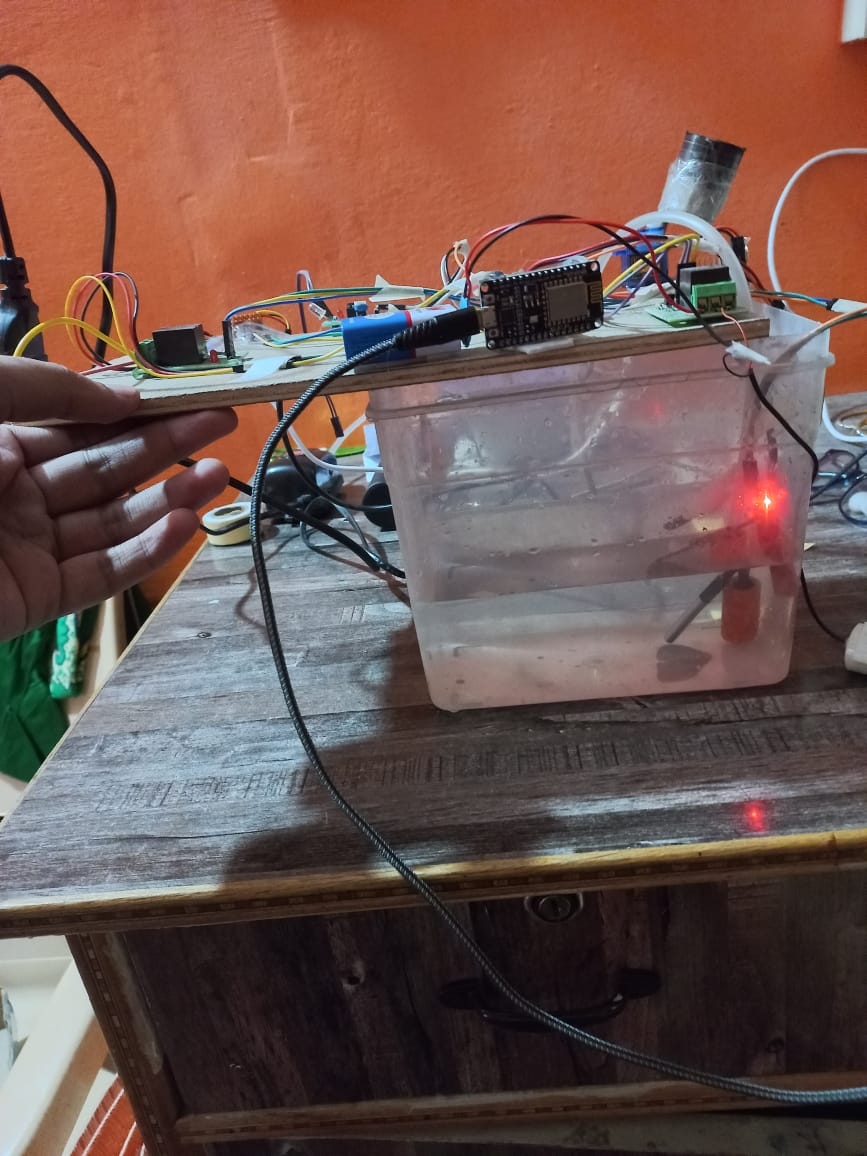
****

Fig.7.2 HardWare Prototype

**Objective:**

To set up a cost-effective, power-efficient, and real-time responsive IoT hardware platform for aquaculture monitoring and control.The hardware setup included an ESP8266 microcontroller, various water quality sensors (pH, TDS, temperature, water level), a servo motor for feeding, a relay module to control oxygen and water pumps, and an ESP32-CAM for visual monitoring. The components were powered using a regulated power supply, and connected in a way that allowed real-time data collection and automation. The hardware was tested in real conditions to ensure accurate readings and smooth operation of motors and the camera. The setup was designed to be cost-effective and suitable for small-scale fish farms, with optional expansion for solar power and edge computing.

**Hardware Requirements:**

Microcontroller: ESP8266 NodeMCU – main controller for sensor and actuator communication.

**Sensors:**

* pH Sensor (4502C)
* TDS Sensor
* Water Temperature Sensor (DS18B20)
* Water Level Sensor

**Actuators:**

* Relay modules for pump and motor control
* Servo motor for feeding
* Oxygen motor
* Inlet and Outlet Motors
* Camera Module: ESP32-CAM for real-time visual monitoring.
* Power Supply: 5V/12V regulated power supply for motors and ESP modules

**Connectivity**: Wi-Fi-enabled for Blynk cloud communication.

**Deployment and Testing:**

* **Sensor Testing:** Verified sensor calibration and accuracy in real pond conditions.
* **Motor Testing:** Confirmed activation logic for oxygen, outlet motors based on sensor thresholds.
* **Servo Testing:** Ensured precise food dispensing using timed servo rotation.
* **Camera Integration:** Tested real-time image capture and transmission upon alert trigger or manual command.

#### **Observations and Challenges:**

* Limited memory on ESP8266 constrained image or camera data processing.
* Variations in water sensor values due to temperature and contamination required calibration.
* Wi-Fi instability occasionally delayed cloud communication.
* Manual calibration and tuning required for motors and servo accuracy.
* Environmental noise (e.g., bubbles, shadows) sometimes misled camera detection.

**CHAPTER 8 TESTING AND RESULTS**

**8.1 RESULTS**

# hard1hard2

Fig:8.1 Hardware Prototype

The hardware prototype, as shown in Fig 8.1, integrates multiple sensors and actuators to ensure seamless aquaculture management. The system monitors key water quality parameters including TDS, temperature, and pH. These sensors continuously send data to the Blynk Cloud, which transmits the information to a mobile application for real-time monitoring. When any parameter exceeds the predefined threshold, the system automatically triggers the corresponding actuator.

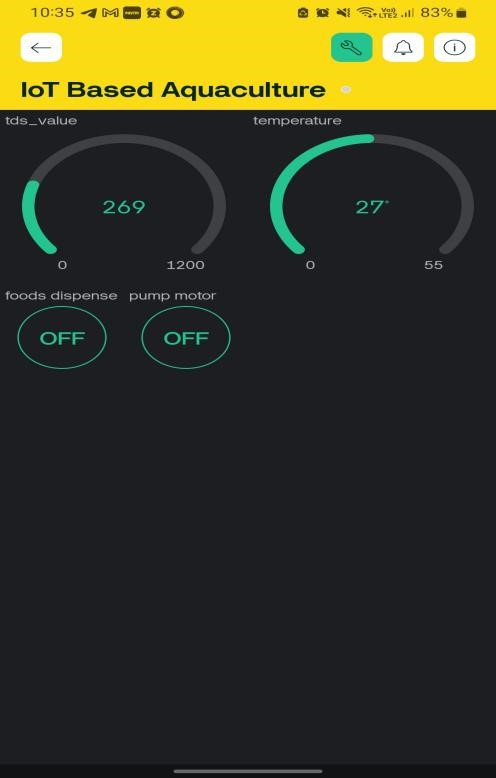


Fig. 8.2 App Dashboard

The figure 8.2 illustrates a simple app page created using the Blynk IoT platform. The interface displays real-time sensor data—including readings for pH, TDS, and temperature—and provides interactive buttons for manual control. These buttons enable the user to trigger automated functions such as feeding and water pump activation, ensuring prompt intervention when water quality parameters deviate from their set thresholds.

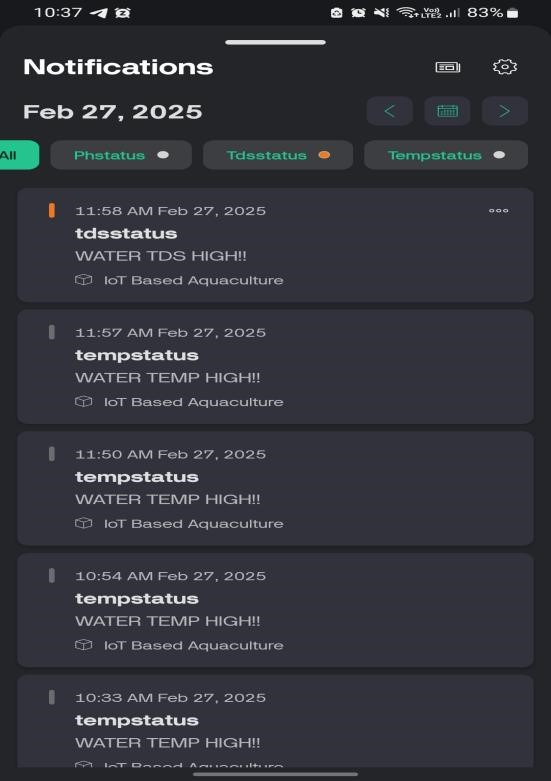


Fig.8.3 Notification Display

This figure 8.3 shows automated notifications sent to the mobile app when sensor readings exceed preset thresholds, enabling timely intervention.



Fig.8.4 Serial monitoring (Data)

This figure 8.4, displays the serial monitoring of real-time sensor data, allowing continuous observation of water quality parameters.

# **CHAPTER 9 Conclusion and Future Work**

**9.1. CONCLUSION**

The iot-enabled smart aquaculture system represents a transformative advancement in contemporary fish farming practices by combining iot sensors cloud-based analytics and mobile connectivity this solution delivers unprecedented monitoring capabilities and automated control the system tracks four vital water parameters in real-time ph levels water temperature dissolved oxygen content total dissolved solids tds through intelligent automation it precisely regulates feeding schedules and aeration processes optimizing aquatic conditions for improved fish welfare and operational sustainability instant notifications and cloud-based analytics enable farm operators to make informed decisions remotely significantly boosting both yield and profitability looking ahead the systems evolution will incorporate sophisticated machine learning algorithms to analyze fish activity patterns derive actionable intelligence from water quality metrics predict potential system issues these enhancements will elevate the platforms analytical capabilities overcoming existing constraints to establish a new standard for proactive data-driven aquaculture management.

**9.2. SCOPE FOR FUTUREWORK**

To enhance the capabilities of the proposed smart aquaculture system, future developments can incorporate advanced technologies as follows:

* **AI/ML-Based Data Threshold Prediction Using Raspberry Pi** Artificial Intelligence (AI) and Machine Learning (ML) algorithms can be deployed on lightweight computing platforms such as the Raspberry Pi to analyze historical sensor data and predict critical threshold deviations. This would allow for proactive interventions, reducing the risk of fish mortality and ensuring sustained optimal conditions in the aquatic environment.
* **Fish Behavior Analysis Using CNN and Camera Feeds** Convolutional Neural Networks (CNNs) can be trained on live video feeds from underwater cameras to detect and classify fish behavior patterns. This can help in identifying anomalies such as stress, sickness, or overfeeding. Automating this behavioral monitoring enhances the precision of feeding schedules and health management, contributing to overall system intelligence and sustainability.

# **Chapter 10 REFERENCES**

[1] S. Saha, R. Hasan Rajib and S. Kabir, "IoT Based Automated Fish Farm Aquaculture Monitoring System," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 2018, pp. 201-206, doi: 10.1109/ICISET.2018.8745543.

[2] S. Tseng, Y. Li and M. Wang, "An application of internet of things on sustainable aquaculture system," 2016 International Conference on Orange Technologies (ICOT), Melbourne, VIC, pp. 17-19, 2016.

[3] M. H. Rohit, Z. T. Hoque, S. M. M. Karim and S. Siddique, "Cost Efficient Automated Pisciculture Assistance System Using Internet of Things (IoT)," 2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT), Montreal, QC, Canada, pp. 49-52, 2019. 4

[4] J. Chen, W. Sung and G. Lin, "Automated Monitoring System for the Fish Farm Aquaculture Environment," 2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, pp. 1161-1166, 2015.

[5] Y. Ma and W. Ding, "Design of Intelligent Monitoring System for Aquaculture Water Dissolved Oxygen," 2018 IEEE 3rd Advanced Information Technology, Electronic and A556utomation Control Conference (IAEAC), Chongqing, pp. 414-418, 2018.

[6] F. L. Valiente et al., "Internet of Things (IOT)-Based Mobile Application for Monitoring of Automated Aquaponics System," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, pp. 1-6, 2018.

[7] Yi-Bing Lin, Hung-Chun Tseng, “Fish Talk: An IoTbased Mini Aquarium System”, 2018 IEEE. Translations and content mining, 2018

[8] K.Raghu Sita Rama Raju, G.Harish kumar Varma, “Knowledge Based Real Time Monitoring System for Aquaculture Using IoT”, 2017 IEEE 7th International Advance Computing Conference, 2017

[9] Sneha P.S, Rakesh V.S, “Automatic Monitoring and Control of Shrim Aquaculture and Paddy Field Based on Embedded System and IoT”, Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017), 2017

[10] Abinaya, J. Ishwarya and M. Maheswari, "A Novel Methodology for Monitoring and Controlling of Water Quality in Aquaculture using Internet of Things (IoT)," 2019 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, Tamil Nadu, India, pp. 1-4, 2019.

[11] K. R. S. R. Raju and G. H. K. Varma, "Knowledge Based Real Time Monitoring System for Aquaculture Using IoT," 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, pp. 318-321, 2017.

## **APPENDIX**

### **A. SENSOR DATA SOURCES**

### The following sensors and sources were used for real-time data collection and monitoring:

* **pH Sensor (4502C)** – Measures the acidity or alkalinity of water.
* **TDS Sensor** – Determines the total dissolved solids present in the water.
* **DS18B20 Temperature Sensor** – Captures real-time water temperature.
* **Water Level Sensor** – Detects the water level in the fish tank or pond.
* **ESP8266 Microcontroller** – Reads data from sensors and sends it to the cloud.

### **B. SOFTWARE TOOLS AND LIBRARIES**

The following tools and libraries were used throughout development:

* **Programming Language:** C++ (Arduino)
* **Cloud Platform:** Blynk IoT Cloud
* **Libraries and Tools:**

**Arduino IDE** – Programming the ESP8266 and other components

**Blynk Library** – IoT device-cloud communication

**Blynk IoT App** – Mobile application used for real-time monitoring, control, and visualization of sensor data through widgets like gauges, graphs, notifications, and buttons.

### **C. SYSTEM COMPONENTS AND ARCHITECTURE**

### **ESP8266 12E** – Acts as the central microcontroller.

### **Sensors (pH, TDS, Temp, Water Level)** – Real-time data collectors.

* **Servo Motor** – Automates fish feeding system.
* **Relay Module** – Controls water pump and oxygen motor.
* **Camera Module** – Captures images during abnormal conditions or manually.

**D. EVALUATION METRICS**

System performance was evaluated based on the following:

* **Latency** – Data update time on the cloud (< 5 seconds)
* **Uptime** – Availability of the monitoring system (target: 99.9%)
* **Accuracy** – Correct detection of abnormal water conditions
* **User Interaction** – Response time between alerts and user actions
* **Power Efficiency** – Battery utilization during offline hours

1. **SAMPLE CODE SNIPPET**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

**char auth[] = "YourBlynkToken"; // Replace with your Blynk Auth Token**

**char ssid[] = "iot"; // Replace with your WiFi SSID**

**char pass[] = "12345678"; // Replace with your WiFi Password**

**int tdsPin = A0; // Analog pin connected to TDS sensor**

**void setup() {**

**Serial.begin(9600);**

**Blynk.begin(auth, ssid, pass);**

**}**

**void loop() {**

**int rawValue = analogRead(tdsPin);**

**// Convert raw analog value to voltage (3.3V reference for ESP8266)**

**float voltage = rawValue \* (3.3 / 1024.0);**

**// Estimate TDS value in ppm (basic approximation for demonstration)**

**float tdsValue = (133.42 \* voltage \* voltage \* voltage**

**- 255.86 \* voltage \* voltage**

**+ 857.39 \* voltage) \* 0.5;**

**Serial.print("TDS Value: ");**

**Serial.print(tdsValue);**

**Serial.println(" ppm");**

**// Send TDS value to Blynk (e.g., Virtual Pin V2)**

**Blynk.virtualWrite(V2, tdsValue);**

**delay(1000);**

**Blynk.run();**

**}**

### **F. FIGURES AND TABLES**

* **Figure 4.1- Figure 4.2** – Sensor Data Flowchart
* **Figure 6.1** – System Architecture Diagram
* **Figure 8.1 -** Hardware Prototype
* **Figure 8.2 -** App Dashboard
* **Figure 8.3** Notification Display
* **Figure 8.4** Serial monitoring (Data)
* **Table 3.1** – Littérature review

### **G. HARDWARE CONFIGURATION**

#### **Primary Development Setup:**

#### **Microcontroller:** ESP8266 12E NodeMCU

#### **Sensors:** pH, TDS, Temperatur, Water Level

#### **Power Supply:** 5V Adapter

#### **Storage & Display:** Blynk Cloud

* **Operating System:** Arduino IDE, Windows 10 / Ubuntu 20.04

#### **Optional Deployment: Edge Device:** Raspberry Pi / Jetson Nano (for image analysis) **Deployment Framework:** MQTT / REST API for cloud sync **Camera Interface:** USB/ESP32-CAM with cloud-triggered capture

**SAMPLE CODE**

#define BLYNK\_TEMPLATE\_ID "TMPL3itBjW2eH"

#define BLYNK\_TEMPLATE\_NAME "IoT Based Aquaculture"

#define BLYNK\_AUTH\_TOKEN "41neT73UmL9gncTDu4q\_URjHJuwLgc2x"

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <Wire.h>

#include <Servo.h>

#include <DallasTemperature.h>

#include <OneWire.h>

#define ONE\_WIRE\_BUS D5 //D1 pin of nodemcu

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire); // Pass the oneWire reference to Dallas Temperature.

Servo myservo;

BlynkTimer timer;

int relay = D8;

int ph = D7;

int phval;

int water =D6;

int waterval;

int pump =D1;

bool tempstatus =false;

bool phstatus =false;

bool tdsstatus =false;

bool waterstatus =false;

float temp;

namespace pin {

const byte tds\_sensor = A0;

}

namespace device {

float aref = 1.0; // ESP8266 ADC range is 0-1V, adjust if using a voltage divider.

}

namespace sensor {

float ec = 0;

unsigned int tds = 0;

float waterTemp = 25.0; // Default water temperature, can use a temperature sensor for actual value.

float ecCalibration = 1.0; // Adjust this value based on calibration.

}

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "iot";

char pass[] = "12345678";

float t;

float h;

BLYNK\_WRITE(V2)

{

int s =param.asInt();

if(s ==1)

{

myservo.write(90);

}

else

{

myservo.write(0);

}

}

BLYNK\_WRITE(V3)

{

int p =param.asInt();

if(p ==1)

{

digitalWrite(pump,HIGH);

}

else

{

digitalWrite(pump,LOW);

}

}

void setup() {

Serial.begin(9600);

sensors.begin();

myservo.attach(D2,500,2400);

myservo.write(0);

pinMode(ph, INPUT);

pinMode(relay, OUTPUT);

pinMode(water,INPUT);

pinMode(pump,OUTPUT);

Blynk.begin(auth, ssid, pass);

timer.setInterval(5000, tmp);

timer.setInterval(1000, sen);

timer.setInterval(60000,up);

}

void loop() {

Blynk.run();

timer.run();

}

void up()

{

Blynk.virtualWrite(V0, sensor::tds);

Blynk.virtualWrite(V1,temp);

}

void sen() {

// Read raw ADC value

float rawVoltage = analogRead(pin::tds\_sensor) \* device::aref / 1024.0;

//Serial.print("Raw Voltage: "); Serial.println(rawVoltage, 2);

// Temperature compensation

float temperatureCoefficient = 1.0 + 0.02 \* (sensor::waterTemp - 25.0);

sensor::ec = (rawVoltage / temperatureCoefficient) \* sensor::ecCalibration;

// Convert EC to TDS

sensor::tds = (133.42 \* pow(sensor::ec, 3) - 255.86 \* pow(sensor::ec, 2) + 857.39 \* sensor::ec) \* 0.5;

Serial.print(F("TDS: ")); Serial.println(sensor::tds);

Serial.print(F("EC: ")); Serial.println(sensor::ec, 2);

// Blynk.virtualWrite(V1, sensor::ec);

if (sensor::tds >= 250 && !tdsstatus)

{

tdsstatus =true;

digitalWrite(relay,HIGH);

Serial.println("Not good");

Blynk.logEvent("tdsstatus","WATER TDS HIGH!!");

}

else if(sensor::tds <250 && tdsstatus)

{

tdsstatus =false;

digitalWrite(relay,LOW);

}

}

void tmp() {

phval = digitalRead(ph);

waterval =digitalRead(water);

Serial.println("Water:"+String(waterval));

sensors.requestTemperatures();

Serial.println(sensors.getTempCByIndex(0));

temp = sensors.getTempCByIndex(0);

Serial.println("Temperature is: ");

Serial.println(temp);

if (temp>= 30 && !tempstatus)

{

tempstatus =true;

Serial.println("TEMP HIGH");

digitalWrite(relay,HIGH);

Blynk.logEvent("tempstatus","WATER TEMP HIGH!!");

}

else if(temp<30 && tempstatus)

{

digitalWrite(relay,LOW);

tempstatus =false;

}

if (phval == 1 && !phstatus) {

phstatus=true;

Blynk.virtualWrite(V8, "WATER PH DETECTED");

Blynk.logEvent("phstatus","WATER pH HIGH!!");

}

else if(phval == 0 && phstatus)

{

phstatus=false;

Blynk.virtualWrite(V8, "NO WATER PH DETECTED");

}

if (waterval == 1 && !waterstatus) {

waterstatus=true;

//digitalWrite(pump,LOW);

Blynk.virtualWrite(V8, "WATER IS high");

Blynk.logEvent("waterstatus","WATER IS high!!");

}

else if(waterval == 0 && waterstatus)

{

waterstatus=false;

//digitalWrite(pump,HIGH);

Blynk.virtualWrite(V8, "WATER DETECTED");

// Blynk.logEvent("waterstatus","WATER IS LOW!!");

Serial.println("Pump on");

}

}

GitHub Link : https://github.com/BasavarajRaviSadargond/Team-15