CYNOSENS: IOT BASED HARMFUL ALGAL BLOOM PREDICTION SYSTEM USING RANDOM FOREST REGRESSION THROUGH WATER QUALITY PARAMETERS

A Project Study Presented to the Faculty of

Electronics Engineering Department

College of Engineering

Technological University of the Philippines

In Partial Fulfillment of the Course Requirements for the Degree of

Bachelor of Science in Electronics Engineering

Submitted by:

Delos Reyes, Vincent J.

Fernando, Ariel Jr S.

Onrada, Verly Ann Jane F.

Sabordo, Louis Edward

Tioco, Carl Vincent V.

Adviser:

Engr. Gilfred Allen M. Madrigal

August 2023

APPROVAL SHEET

This project entitled "**IoT BASED HARMFUL ALGAL BLOOM PREDICTION SYSTEM USING RANDOM FOREST REGRESSION THROUGH WATER QUALITY PARAMETERS"** has been approved and submitted by the following proponents:

Delos Reyes,Vincent J.

Fernando, Ariel Jr. S.

Onrada, Verly Ann Jane F.

Sabordo, Louis Edward

Tioco, Carl Vincent V.

In partial fulfillment of the requirements for the degree of **Bachelor of Science in Electronics Engineering** is hereby recommended for approval:

ENGR. GILFRED ALLEN M. MADRIGAL
Project Adviser

ENGR. LEJAN ALFRED ENRIQUEZ
Panel Member

ENGR. EDGAR A. GALIDO
Panel Member

ENGR. ROMEO L. JORDA JR.
Panel Member

ENGR. CHERRY G. PASCION
Panel Member

ENGR. JOHN PETER M. RAMOS
Panel Chair

Accepted and approved in partial fulfillment of the course requirements for the degree of **Bachelor of Science in Electronics Engineering.**

ENGR. TIMOTHY M. AMADO
Head, ECE Department

ENGR. NILO M. ARAGO
Dean, College of Engineering

ACKNOWLEDGEMENT

The proponents would like to express their deepest appreciation and gratitude to all the individuals and organizations who have contributed to the successful completion of this thesis. This research endeavor would not have been possible without their invaluable support, guidance, and encouragement. The proponents would like to extend their sincere thanks to the following:

First, the proponents would like to thank the God Almighty, for making this possible. The proponents thank His almightiness for giving us the wisdom, love, and guidance to power through this paper. To God be the Glory!

To **Engr. Gilfred Allen M. Madrigal**, for his unwavering support, expertise, and invaluable guidance throughout this research project. His immense knowledge and insightful feedback have been instrumental in shaping this thesis and ensuring its academic rigor. The commitment and efforts he exerted will always be remembered.

To the panel members, **Engr. Lejan Alfred C. Enriquez, Engr. Edgar A. Galido, Engr. Romeo L. Jorda Jr., Engr. Cherry G. Pascion, and Engr. John Peter M. Ramos**, for providing the study with a conducive research environment and access to resources essential for the successful execution of this study. The academic excellence and commitment to research exhibited by the institution have played a crucial role in enhancing the quality of this thesis

To the **Bureau of Fisheries and Aquatic Resources 4A - Batangas Inland Fisheries Technology Outreach Station and Victoria Corrales** (Fishpond Owner in Lawa ng Laguna, Brgy. Malaban, Biñan, Laguna), for assisting the proponents in selecting the appropriate deployment site of the study. A special thanks to Ms. Shaira C. Javier, RCh (Aquacultural Technologist) and Mr. John Alfred J. Palcon (Aquaculturist I) for being present during the deployment process and providing ideas and advice that could help the thesis project.

To **Mr. Johan Nikolai C. Nepomuceno**, for his assistance and service to improve the project study.

To **Department of Science and Technology-Science Education Institute Undergraduate Scholarship**, for the support of the government in providing financial aid for the development of science and the knowledge of the country and to increase the interest of Filipino students in partaking in researchers for the country, the proponents thank you for your support, as without the program, the paper would not have been presented to the panel and the people, the proponents sincerest gratitude towards your aid.

To the **Onrada family** for allowing the proponents to conduct the project study in the comfort of their home. They will always be remembered for their friendly reception and hospitality to the proponents.

To the **families and friends** of the proponents, for their unwavering encouragement, understanding, and patience throughout the duration of this research. Their unconditional support has been a constant source of motivation, enabling the proponents to overcome challenges and strive for excellence.

ABSTRACT

One of the various economic uses of different bodies of water is aquaculture; however, one of the major problems it deals with nowadays is fish kills that are caused by fluctuations in the natural environment that leads to eutrophication — a process that results in the accumulation of harmful algal blooms (HABs). HABs are dangerous for aquatic systems as they slow the growth of fish and other aquatic animals. Thereby, the aim of the study was to develop a standalone monitoring device that is efficient and a real-time system to detect the presence of algal blooms and predict when they will become harmful to aquatic animals. Specifically, the monitoring buoy is equipped with six water quality sensors for the following parameters: pH, Temperature, Conductivity, Dissolved Oxygen, Nitrate, and phosphate, utilizing Random Forest Regression as a predictive algorithm and the Internet of Things. The monitoring device is connected to the web application via a wireless network, and the prediction function of the device is based on the time at which it was brought into the water to undergo a quality test. The web app displays a warning that alerts users to the presence of HABs and whether certain measured values are already detrimental to the fish. The system triggered at a 21 μg/L level of algae in the water and detected harmful algae content greater than 50 μg/L with 91.48% prediction model accuracy, which is beneficial as it has the capability to determine an extensive range of HAB levels in the water. Results demonstrated that HAB monitoring buoys could rapidly and efficiently supplement manual sampling in at-risk water bodies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

THE PROBLEM AND ITS BACKGROUND

This chapter provides an overview of the topics essential to a deeper understanding of the study's concepts. This chapter includes the introduction, background of the study, research gap, research objectives, significance of the study, scope and limitations, and definition of the terms.

1.1 Introduction

The Philippines is fortunate to have such an abundance of natural resources; one endowed resource is water. The aquatic biome of the country is classified into two categories: freshwater and marine regions. Rivers, lakes, and groundwater are examples of inland freshwater, whereas bay, coastal, and oceanic waters are examples of marine water. These bodies of water are vital to the country's economic development since they support the cultivation of various crops that meet an individual's demands. One economic use of diverse bodies of water in the Philippines is aquaculture; it involves breeding, raising, and harvesting various fish species and aquatic plants. Likewise, rapid aquaculture growth has arisen since the discovery that different species of fish, including those not native to its water, can be grown in controlled environments in the lake. Nowadays, in spite of its importance, due to rapid population growth, urbanization, industrialization, and a lack of access to safe and sufficient water, the quality of Philippine waters is diminishing, a serious problem facing the country today. Wastewater discharged from domestic, industrial, and agricultural runoff contains detergents, fertilizers, heavy metals, chemicals, oils, and even solid waste materials, seriously damaging the recipient water bodies. Each of these pollutants has a distinctive toxic repercussion that affects human livelihood, resulting in economic costs.

In line with the detrimental activities that affect the aquaculture of the Philippines, several problems are being encountered by fishermen in different regions, and the most common problem is fish kill, which occurs because of human activities and natural causes. Fishkill, as defined by the Bureau of Fisheries and Aquatic Resources (BFAR), is the "massive death of fish stocks in a specific area as a result of unfavorable water quality parameters of a given aquatic environment that are intolerable or toxic to the fish stocks." Additionally, the body of water is composed of a number of parameters namely: pH level, turbidity, conductivity, temperature, salinity, and many others that are vital in perceiving and understanding the quality of the given water. Each parameter has a different degree of importance depending on the purpose of study that has to be done in the water.

The formation of harmful algal blooms (HAB) is one of the natural causes when not treated immediately, could lead to fish kill. HAB is the rapid enrichment of algae or cyanobacteria that occurs on the surface of lakes and other bodies of water when the nutrients on which algae depend are overabundance in water. Thus, the unfavorable water quality parameters cause the algae to bloom in the water. Moreover, algal blooms are harmful because they reduce fish and other aquatic life's ability to find food, block sunlight from underwater plants and consume oxygen — the insufficient oxygen makes it impossible for aquatic life to survive. On this note, it is highly relevant to predict the bloom of harmful algae to protect the aquatic life and for the fisherfolk to know when to take necessary action to prevent it.

There are six water quality parameters responsible for harmful algal bloom occurrence in water namely: pH level, nitrate, phosphorus, temperature, electrical conductivity, and dissolved oxygen. Similarly, low levels of dissolved oxygen, ammonia, nitrite, or hydrogen sulfide in shrimp and fish cause stress and illness. Ammonia and hydrogen sulfide toxicity can be increased by temperature and pH abnormalities. Moreover, excess nitrogen and phosphorus result in algae growth (Prananingtyas et al., 2019). Hence, having appropriate amounts of these water quality indicators plays an important role in preventing harmful algal formation.

With the advancement of technology, various tools, and sensors were developed to monitor water quality parameters. These include the use of electrochemical and optical sensors, among many others. Algal bloom monitoring is done by water sampling, and the parameters are being studied in laboratories. In addition, a water station buoy was developed to remotely monitor the water parameters to know the quality of a given water, however, most of these devices are imported from other countries and are particularly expensive.

The progress in developing computing capabilities has brought modern techniques to predict the harmful algal bloom in water. In-depth literature review encompassing different applied machine learning techniques models for predicting harmful algal bloom, artificial neural network (ANN) and support vector machine (SVM) are two commonly used. Conversely, the experimental results show that the improved backpropagation (BP) algorithm and SVM perform better than (generalized regression neural network) GRNN methods. Even so, analysts still recommend that these algorithms still need to be further improved.

In this study, the algorithm employed in machine learning in a certain data set is by using Random Forest Regression. Random Forest is a versatile and easy-to-use machine learning algorithm that consistently delivers good results without hyperparameter adjustment. It is also one of the most extensively used algorithms due to its simplicity and can be used for classification and regression problems. The "forest" it generates is a collection of decision trees trained using the "bagging" method. Bagging techniques are based on the concept that combining learning models enhances overall performance.

All of this has encouraged the researchers to develop a standalone prediction system that predicts the presence of HABs in bodies of water by monitoring water quality indices. The water sensors used in the device were dissolved oxygen, water temperature, conductivity, nitrate, phosphate, and pH level. The prediction system was programmed using Random Forest Regression, and the data collected by the device was displayed in a web application.

1.2 Background of the Study

Water is recognized as one of the scarcest natural resources on the planet and is essential for the emergence of life. However, nowadays, as the population increases and the development of technologies continues to foster, its impact on the environment becomes more noticeable. The current increase in environmental degradation has a particular impact on water sources. Today, water can either be a source of life and good health or a source of diseases and deaths, depending on its quality. Due to this fact, in order to monitor remote rivers, lakes, coastal areas, and other water bodies, a study entitled A System for Monitoring Water Quality in a Large Aquatic Area Using Wireless Sensor Network Technology, created a low-cost, real-time system. The system is designed to monitor the three water parameters of dissolved oxygen (DO), pH, and temperature. In addition, it detects each of these parameters at predetermined intervals of time. To better serve concerned end users, the built prototype disseminates the obtained data in graphical and tabular representations via a dedicated web-based portal and preregistered mobile phones. In order to reliably monitor water quality indicators, the system used readily available electrochemical sensors. It then used NodeMCU and pocket wifi to present the data online. Sent data from the sink node to the base station was transmitted using it, the NodeMCU, and pocket wifi (Hongpin et al., 2015).

On the study conducted by Lorenzo et al. (2019), the researchers created a multi-function, low-cost system and used Decision Trees, Decision Forests, and Multi-layer Perceptron machine learning algorithms to hydrologically model water parameters of temperature, pH, conductivity, and calculate dissolved oxygen (DO) values. The Random Forest algorithm gave the most accurate measurements when compared to the other two algorithms, hence it was used to build the most effective model employing a variety of metrics. The instrument has a 2.61% error margin when predicting a specific water pond's dissolved oxygen content compared to Atlas Scientific's DO Sensor. The developed device is handheld and made up of sensors for the factors that influence DO the most, namely temperature, conductivity, and pH.

Torres et al. (2020) established a wireless network-based device for real-time monitoring to keep track of many aspects of the water quality for aquaculture in order to enhance the quality of the products produced, address wiring issues, and create a more affordable monitoring system. The monitoring system is powered by solar cells and lithium cells to make the device self-sustaining. The water's properties were tracked using various sensors, including oxygen, pH, temperature, nitrogen, and ammonia sensors. Zigbee and GPRS modules were utilized to send data collected to the monitoring center; a STM32F103 chip was used for data processing. The test results demonstrated high transmission reliability with a packet loss rate of only 0.43%. With a packet loss rate of 0.43%, the data acquired after the device was tested demonstrated good data transmission confidence. To conclude, the system is reliable for transmitting information for remote, real-time aquaculture water quality monitoring.

Due to hydrology and other factors including the changing seasons of biological processes, stream nutrient concentrations show significant temporal variation. For correct estimation of nutrient loadings and to completely define lotic nutrient conditions, many water quality monitoring programs sample too rarely (i.e., weekly or monthly).The surrogate-regression strategy, which estimates nutrient concentrations from related measures (such as conductivity or turbidity) that can be easily detected in situ at high frequency using sensors, is a common solution to this issue. However, skewed distributions, nonlinear correlations, and multicollinearity are frequently present in stream water quality data, which might provide challenges for linear-regression models. In this study, the researchers estimated stream nitrogen (N) and phosphorus (P) concentrations through sensor data inside a forested, mountainous drainage region in upstate New York using the adaptive and reliable machine learning technique Random Forests Regression (RFR). This method demonstrated variations in nitrate (89%), total nitrogen (85%), particulate phosphorus (76%), and total phosphorus (74%), when compared to real nutrient data from laboratory-tested samples. Even though the values of these later parameters were in a very low range, the models' predictions for total soluble P (47%) and soluble reactive P (32%) were less accurate. Despite not being frequently utilized as substitutes in nutrient-regression models, soil moisture and fluorescent dissolved organic matter were key indicators in this study. Researchers have come to the conclusion that RFR has a lot of potential as a tool for predicting instantaneous stream nutrient concentrations from high-frequency sensor data and therefore invite others to test this method as an addition to conventional (laboratory-determined) nutrient datasets (Prananingtyas et al., 2019).

Diverse kinds of microalgae commonly thrive and proliferate in various aquaculture ponds in the provinces of Bulacan, Zambales , Tarlac, Nueva Ecija, and Pampanga. Meanwhile, some of these microalgae species have negative environmental effects and an adverse impact on the aquaculture sector. After identifying and characterizing the composition of microalgae that can form HABs (Harmful Algal Blooms) using morphological, ultrastructural, and molecular characterization, potential HAB formers such as Oscillatoria agardhii, O. princeps, Microcystis aeruginosa, and M. wesenbergii were observed to have occurred mostly, while Anabaena circinalis was only observed in one site. Different physicochemical properties of pond water, especially nutrient levels such as nitrate, and orthophosphate have also influenced the occurrence of these cyanobacteria (Harrison et al., 2021).

1.3 Research Gap

Contingent on associated research, improvements are needed to practically and efficiently address the lacking factors and problems entail to be answered. To begin with, most of them utilized, at maximum, three parameters only in monitoring and testing the quality of water. Specifically, pH, dissolved oxygen, and salinity. While others cover only Nitrate and Phosphate parameters in their study.

Meanwhile, the incorporation of predicting the occurrence of Algal Bloom in different bodies of water can be beneficial to a water quality meter/monitoring device as this is one of the leading causes of fish kill and poor water attributes. Developing better feedback mechanisms is also essential to easily determine the given data brought by the system.

1.4 Research Objectives

The main goal of this study was to develop a real-time aquatic device that can predict the Harmful Algal Blooms in water through the utilization of water quality parameter sensors in a wireless sensor network via machine learning algorithms.

Specifically, the study aimed:

1. To develop a standalone algal bloom monitoring buoy utilizing water quality sensors for dissolved oxygen, temperature, pH level, nitrate, phosphorus, and conductivity;
2. To develop a program to predict the harmful algal bloom using Random Forest Regression;
3. To develop a website application to display the acquired data and;
4. Test the device functionality through simulations and actual field testing, and evaluate the accuracy by comparing these measurements to an actual algal bloom monitoring device.

1.5 Significance of the Study

The Philippines is an archipelago comprising approximately 7,107 islands with different individuals who appreciate the vastness of the surrounding waters and seas. As a result, aquaculture is one of the primary sources of income for most Filipinos. One of its significant waters is the Laguna Lake, which is a significant fishing location, but it also serves many other purposes, among its other uses is a supply of water for agriculture, electricity generation, and industrial cooling activities.

The BFAR defines fish kill as the mass death of fish stocks in a certain area due to toxic water quality parameters in an aquatic environment. One common cause is algal blooms and the resulting water quality issues, such as low oxygen and toxin production.

The present study sought to develop a device with a web application for monitoring the algal bloom through water quality. With the application of the latest technology in the field of

Agriculture, Aquatic, and Natural Resources Sector (AANR) which comes under the Harmonized National Research and Development Agenda's (HNRDA): Agriculture: Fisheries and Aquaculture categories.

Also, the Sustainable Development Goal (SDG), whose goal was to create a set of universal goals to aid in the fight against the world's severe environmental, political, and economic concerns. This study was undertaken under SDG 6: Ensure availability and sustainable management of water and sanitation for all and SDG 12: Ensure sustainable consumption and production patterns.

Lastly, this study would have a socioeconomic impact on community safety by providing a device to guarantee the pond's fish is safe and fresh.

1.6 Scope and Limitations

The study focused on the development of a real-time aquatic device that can predict the occurrence of harmful algal bloom. Those parameters include the Dissolve Oxygen (DO), Temperature, pH level, Nitrate, Phosphorus, and Electrical Conductivity. Moreover, the data collected by the device was seen and scrutinized through website application via wireless network connection. Multiple sensors are built inside the standalone device to capture various pond water parameters.

Considering that Algal Bloom occurrence is dependent on weather patterns of a specific area, the prediction function of the device was based on the time it will be brought into the water to undergo a quality test. There will be a warning that will notify the users about the occurrence of harmful algal bloom.

The study does not cover other different parameters such as Carbon Dioxide (CO2), Biochemical Oxygen Demand (BOD), Hydrogen Sulfide (H2S), and Alkalinity. The water quality and prediction of Algae will focus on the overgrowth or 'bloom' of algae through the use of water quality parameter sensors. Prevention of harmful Algal Bloom is not concerned under the scope of the study.

Even though, theoretically, it is possible for the device to predict the HAB in any body of water given enough training samples, this study collected data only from small-scale aquatic resources, specifically a fishpond, in order to have an easier access to reference water parameters values used as training, testing, and validating.

1.7 Definition of Terms

To better grasp the conducted study, the following terms were defined based on the context of the research.

Algal Bloom - a sudden rise in the number of algae or their accumulation in freshwater or oceanic water systems.

Aquaculture - This is the process of controlled cultivation of aquatic organisms for breeding, rearing, and harvesting fish and other organisms in all types of water environments.

Dissolve Oxygen (DO) - is one of the essential water quality parameters to be monitored for the survival of fish and other aquatic organisms which determines the amount of oxygen that is present in water.

Electrical Conductivity - the ability of a water sample to conduct an electric current.

Eutrophication - This is the process in which a water body becomes overly enriched with nutrients, leading to plentiful growth of aquatic organisms such as algae that often results in the deterioration of water quality and depletion of dissolved oxygen in water bodies.

Fish Kill - is an instance in which dead or dying fish have been noticed in a lake or body of water.

Internet of Things - is a network of physical objects that are equipped with sensors, software, and other technologies for online communication and data exchange.

Nitrate - refers to the nitrogen fraction of a sample's total nitrate. It is a nitrogen and oxygen-containing chemical molecule and is a vital component for plant growth.

pH level - one of the parameters to be measured that determines the measure of the water's acidity (hydrogen ions) or alkalinity.

Phosphorus - also known as the "limiting nutrient" in the aquatic environment- refers to the fact that the availability of this mineral limits how quickly algae and aquatic plants may grow.

Sensors - pertains to the electronic devices to be used which detect or measure a physical property/quantity and convert it into a signal which an instrument and observers can read.

Temperature - one of the variables that should be watched since it has an impact on consumption patterns, ammonia toxicity, and fish and shrimp metabolism.

Chapter 2

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter presents the synthesis of related literature and studies for the researchers to gain an in-depth understanding of the relevant information. Hence, this chapter established the foundations of the concepts used, and how they will be used in the context of the study.

2.1 Aquaculture

Aquaculture is the controlled cultivation of aquatic organisms such as fishponds and fish pens. According to Demetillo et al., (2019), aquaculture covers almost 50% of fishes on the planet that is being consumed by humans and is expected to increase in the following years. Lorenzo et al. (2019) also said that the aquaculture sector around the globe is continuously growing in the past 40 years, though comparatively the aquaculture industry in some countries has grown rapidly than others. In connection with the large contribution of aquaculture to the world's economy, various factors are being considered such as its contribution to pollution, disease management, antibiotic use, food safety, and access to water, etc.

2.1.1. State of Aquaculture in the Philippines

The Philippines archipelago is surrounded by different bodies of water. With more than 7000 islands, various bodies of water are also being utilized for aquaculture (Anderson et al., 2019). Aquaculture is one of the Philippines' main sources of food; It has been providing livelihood to different locals residing near lakes and seaside. However, given that the country is highly dependent on fish products, aquaculture in the Philippines has been declining because of the decrease in seaweed production (Nadarajah & Flaaten, 2017). According to the Philippine Statistics Authority, the Philippines' aquaculture provided 2,322.91 metric tons of fish products in 2020 and 2,246.32 metric tons in 2021. These numbers show that there is a decrease of 3.3% in the yearly production of fish products in the Philippines (Borlaza et al., 2023).

2.1.2. Fish kills in the Philippines

Alongside the benefits and different uses of water bodies in the Philippines and aquaculture, problems usually arise, especially fish kills, which can be attributed to different causes. In Taal Lake, fish kills were caused by isopod infestation, sulfur upwelling, abnormal temperatures, anomalies in dissolved oxygen levels, and other physico-chemical parameters (White, 2017). While on the other side of CALABARZON, algal blooms have been occurring periodically in Laguna Lake. Cyanobacteria cause HABs or Harmful Algal Blooms that consume oxygen in the lake, leading to the death of fish caused by oxygen depletion (Mapa, 2022).

In May 2019, a fish kill incident occurred in Biñan and Pila, Laguna, as thousands of dead tilapias and bangus were scattered in the fishing zone. An estimated 500,000 fish have died while the remaining produce foul-odor, which is no longer profitable. Ireneo G. Bongco, Senior Science Research Specialist at Laguna Lake Development Authority (LLDA) said that the fish kill was caused by a green tide (red tide in saltwater) or algal blooms, where the fish suffered from asphyxia due to deficiency in dissolved oxygen (DO) level concentration brought by the competing thick layer of green tide in the lake surface. These fish kills caused by oxygen deficiency always damage the fishermen's livelihood around the lake (Torres, 2020).

Additionally, on April 18, 2022, the public in Bolinao, Pangasinan, was advised to refrain from consuming all types of shellfish harvested from coastal waters due to the presence of algae that produces the red tide toxin (Mendoza et al., 2019).

Table 2.1 Synthesis for Fish Kills in the Philippines

Author	Year	Title	Relevant Findings	Relationship to the Study
Anderson J., Asche F., and Garlock T.	2019	Economics of Aquaculture Policy and Regulation	Aquaculture has a big contribution to the world's economy and its contribution is continuously growing.	The study focuses on factors affecting the habitat of fish because the occurrence of fish kills has an impact on the economy.
Nadarajah S., and Flaaten O.	2017	Global Aquaculture growth and institutional quality	Global Aquaculture industry has been continuously growing.	Monitoring fish habitat is vital to the aquaculture industry.
Mapa D.	2022	Fisheries Situation Report for Major Species	Aquaculture in the The Philippines has provided an annual amount of approximately 2000 metric tons of fish but the annual harvest has a decline rate of 3%.	Decline in the amount of fish harvested could be the result of fish kills occurring and poor habitat management.
Tiongson K., Tamayo-Zafar alla M.	2018	Water Quality and Seasonal Dynamics of Phytoplankton and Zooplankton in the West Bay of Laguna de Bay, Philippines	Harmful Algal Blooms consume dissolved oxygen in water.	Competition to Dissolved Oxygen leads to fish kill. Monitoring Dissolved Oxygen levels may prevent fish kills.

Visperas E.	2022	Red Tide Warning Up in Pangasinan	Due to the presence of algae that creates the red tide toxin, all shellfish taken near ocean areas are unsafe.	The possible location of the deployment of device

2.2 Algal Blooms

Harmful algal blooms (HABs), sometimes known as "red tides," are a problem in almost all coastal nations worldwide. These occurrences are brought on by blooms of minuscule algae. Some of these algae are harmful and exposure to them can cause disease and even death in people, marine mammals, seabirds, fish, and other aquatic life. Toxins are often spread through the food chain.The direct discharge of harmful substances can occasionally kill marine life. Due to the vast amount of algae collected, non-toxic HABs are often harmful to ecosystems, fish stocks and recreational facilities. The term "HAB" also refers to non-toxic macroalgal blooms (seaweeds), which can have significant ecological effects such as displacing native species, changing habitats, and depleting bottom-water oxygen (Mendoza et al., 2019)

On the other hand, Freshwater Microcystis is a global genus and has developed algal blooms in Laguna de Bay since the early 1970s (Tiongson & Tamayo-Zafaralla, 2018). Algal blooms are a natural occurrence. However, it is one of the nuisances in coastal areas (Anderson, 2009). Blooms are dense clumps of cyanobacterial cells that form in marine, brackish, and freshwater environments that are usually noticeable by the visible discoloration they cause in the water (Espaldon et al., 2005). Algal blooms caused by high fertilizer concentrations also lead to huge changes in dissolved oxygen (DO) concentrations. Large volumes of dissolved oxygen are present in surface waters throughout the daytime during blooms as a result of photosynthesis.

However, at night, hypoxia or anoxia brought on by algal cell respiration could result in fish deaths (U.S. EPA, 2023).

2.2.1 Eutrophication

Water eutrophication is one of the environmental issues we faced until now. Most freshwater lakes are encountering water quality problems and ecological imbalances caused by anthropogenic activities (Heisler et al., 2008). Specifically, nitrogen and phosphorus eutrophication in marine ecosystems. It has a negative impact on food security, ecosystem health and the economy through disruption in fisheries. Uncontrolled levels of nitrogen and phosphorus cause algae blooms, anoxic conditions, and ocean acidification with these conditions leading to dead zones and fish kills (D'Silva et al., 2012). Moving forward, lakes grow more fertile and shallow as a result of eutrophication, which is the process of lakes receiving phosphorus and nitrogen as nutrients and silt from the nearby watershed. The more nutrient-rich a lake is, the more water organisms it can support (D'Silva et al., 2012).

Different factors such as agricultural, municipal, and industrial activities have significantly increased aquatic nitrogen and phosphorus pollution that resulted in the water becoming fertile, known as eutrophication, thus endangering water quality and biotic integrity, from headwater streams to coastal locations. The excessive nutrients in the water contribute to oxygen depletion, ecosystem diversity, and eutrophication (Ngatia et al., 2019) Consequently, Hypoxic "dead zones" reduce fish and shellfish production, harmful algal blooms cause taste and odor problems and threaten the safety of drinking water and aquatic food supplies; greenhouse gas release stimulation; and degradation of cultural and social values of these waters are all problems caused by eutrophication (RMBEL, 2018).

2.2.2 Monitoring of Algal Blooms

Throughout the lifespan of a desalination plant, water quality monitoring is essential for identifying occurrences that result in poor water quality, such as harmful algal blooms (HABs). The organic and solids load in the seawater feed to be treated at a desalination plant might significantly rise as a result of HABs. The crucial for process control and contractual objectives is to monitor the water quality of the intake feedwater to optimize pretreatment procedures in reaction to a decline in feedwater quality brought on by situations such as HAB to sustain both production and quality targets (Oracle Philippines, n.d.).

The suggested model for chlorophyll-a concentration has three primary novelties. In order to explicitly address the complex spatio-temporal dependence in chlorophyll-a concentration on a monthly scale, the MHMM can be used as a framework using the latent (or hidden) classes, which correspond to the mixture components for each observation and are supposed to follow a Markov process. A multivariate hidden Markov model (MHMM) based on gamma mixture distribution was suggested to achieve these goals and construct a decision-support framework for algal bloom at several stations in the lower Nakdong River. The study intends to investigate the spatio-temporal dynamics of phytoplankton dispersion with chlorophyll-a concentration and perhaps cluster them using a limited number of latent states, which can be connected to other water quality measures (Wilhelm, 2009).

2.2.3 Prediction of Algal Blooms

Each year, harmful algal blooms cause environmental harm, financial losses, and outbreaks of diseases. Since there is yet no fundamental resolve to this concern, improving early warnings (predictions) is the best approach to mitigate the effects of algal blooms. Although numerous variable data sources are needed for the study, current physical prediction models struggle to provide a precise coefficient demonstrating the relationship between each aspect when predicting algal blooms. High time and financial costs come along with these constraints. While this is happening, artificial intelligence and deep learning techniques are becoming more and more prevalent in scientific research. As a result, more and more long short-term memory (LSTM) model applications to environmental research challenges are being made because the LSTM model performs well for time-series data prediction. Few studies, particularly in South Korea where annual algal blooms occur, have used deep learning models like LSTM to predict algal blooms. As a result, the researchers used the LSTM model to predict algal blooms in South Korea's four main rivers. Researchers used regression analysis and deep learning approaches to make short-term (one-week) predictions on a recently created water quality and quantity dataset derived from 16 damned pools on the rivers. Chlorophyll-a, a well-known proxy for algal activity, was predicted using three deep learning models (multilayer perceptron, MLP; recurrent neural network, RNN; and long short-term memory, LSTM). The outcomes were contrasted with actual data based on the root mean square error (RSME) and OLS (ordinary least square) regression analysis. The LSTM model generates the highest predictive rate for harmful algae outbreaks, and the OLS regression analysis was outperformed by all deep learning models. The findings demonstrated the potential of LSTM and deep learning for algal bloom prediction (Wurtsbaugh et al., 2019).

The researchers developed several machine-learning-based predictors including AdaBoost (Adaptive Boosting), ANN (Artificial Neural Network), GBDT (Gradient Boosting Decision Tree), KNN (K-nearest Neighbor), and SVM (Support Vector Machine) for model selection in order to improve the accuracy of algal bloom predictions. Moreover, the researchers used an extensive testing procedure to evaluate the performance of all feature subsets under these predictors in order to identify the key environmental variables that influence prediction accuracy. Researchers discovered that the GBDT performs better on both datasets when using particular input combinations after empirical training (Yu et al., 2021).

Table 2.2 Synthesis for Prediction of Algal Blooms

Author	Year	Title	Relevant Findings	Relationship to the Study
Ngatia, L., Grace III, J. M., Moriasi,D., and Taylor, R.	2018	Nitrogen and Phosphorus Eutrophication in Marine Ecosystems	Eutrophication is a major problem in aquatic ecosystems that are driven primarily by nitrogen and phosphorus.	Nitrogen and phosphorus are one of the parameters that the study focuses on.
Tsang, Y. F., and Kwon, H.-H.	2020	Stochastic Modeling of Chlorophyll-a for Probabilistic Assessment and 2 Monitoring of Algae Blooms in the Lower Nakdong River, South Korea	Phytoplankton abundance is generally indicated by chlorophyll-a concentration.	Chlorophyll-a is also present in algal blooms aside from cyanobacteria. Factors affecting the growth of cyanobacteria also affect the increase in chlorophyll-a.

Lee, S., and Lee, D.	2018	Improved Prediction of Harmful Algal Blooms in Four Major South Korea's Rivers Using Deep Learning Models	The annual occurrence of harmful algal blooms causes environmental damage, economic losses, and disease epidemics.	Both studies predict the bloom of algal however, the present study uses Random Forest Regression algorithm.
Anderson, D. M.; Boerlage, S. F. E. and Dixon, M.B.	2017	Harmful Algal Blooms (HABs) and Desalination: A Guide to Impacts, Monitoring and Management.	Given the patterns of population, agriculture, development, and climate around the world, an increase in hazardous algal occurrences is unavoidable.	The study focuses on monitoring and management of Harmful Algal Blooms
Yu, P., Gao, R., Zhang, D., and Liu, Z.P.	2021	Predicting coastal algal blooms with environmental factors by machine learning methods	A machine learning-based approach that uses environmental factors to forecast the presence of algal blooms	The significance of each feature for the execution of the prediction indicates key elements for the occurrence of dangerous algal blooms.

2.3 Water Quality

One of the most important aspects of human life is water. It is essential to plants' and animals' survival and long-term viability. However, today, it is being widely exploited by the dumping of human garbage, industrial waste, and chemical waste into the water bodies, which contaminates the water with harmful compounds. Water scarcity and pollution have been severe problems in many parts of the world. There is a lot of emphasis on industrialization and urbanization in many emerging countries, like India. This shows that water usage has increased and quality has degraded (Kim et al., 2020).

Table 2.3 Synthesis for Water Quality

Author	Year	Title	Relevant Findings	Relationship to the Study
G. S. Menon, M. V. Ramesh, and P. Divya	2017	A low-cost wireless sensor network for water quality monitoring in natural water bodies	The evaluation of water quality is a crucial component of environmental monitoring. When water quality is poor, not only does it damage aquatic life, but also the environment in its vicinity.	Both studies deal with the monitoring of water quality however, the present study is composed of 7 parameters.

2.4 Water Parameter Sensors

Water is a necessary element for fish to survive. Due to the wide range of applications, the measurement of predetermined parameters provides users with real-time online monitoring feedback. Moreover, to monitor the water quality of lakes and rivers and to determine the reasons and factors leading to problems with water quality, the device was equipped with a variety of sensors, a communication link, storage and processing capabilities, and energy for powering and utilization of the device (Lee & Lee, 2018).

2.4.1 Dissolved Oxygen (DO) Sensor

The amount of dissolved oxygen (DO) in a body of water is dramatically affected by algae bloom levels. The amount of DO in the water indicates whether the ecosystem has enough oxygen and whether the marine environment is suitable for aquatic organisms' healthy survival. Dissolve Oxygen sensors are commonly used to assess water quality (Menon et al., 2017). Dissolved oxygen sensor measurements are used to track operations where oxygen content influences reaction rates, process efficiency, or environmental conditions. This sensor is commonly utilized in situations where keeping a constant oxygen level is critical for maintaining or minimizing reactions (Hong et al., 2021).

2.4.2 pH Level Sensor

One of the most important markers of water quality is pH level. The pH parameters are difficult to measure precisely, since the pH level deals with very small amounts of ionic concentration, necessitating the use of a sensitive sensing instrument. A pH sensor is used to offer a general evaluation of water quality. However, the number of sensors can be increased whenever required (Hongpin et al., 2015). The lowest pH value that causes fish death is 4, whereas the maximum is 11. As a result, since it is used to detect the pH value of water, the SEN0161 pH sensor is one of the components in this system. The pH of the water is monitored using these sensors, which are installed at each node (Lauguico et al., 2020).

2.4.3 Temperature Sensor

The water temperature affects fish's appetite and metabolism, since it also affects breeding. Moreover, the DS18B20 digital water temperature sensor application is designed and implemented using Python running on the Rpi core controller to send the real-time water temperature read from the sensor. The digital temperature sensor is used to capture the temperature on the Rpi (Zhang et al., 2020). The temperature values of the Dallas Semiconductors DS18B20 are 9 to 12-bit. The sensor communicates using a one-wire interface, connecting several sensors to a single port. The temperature ranges from -55°C to 125°C, with a +/-0.5°C accuracy (Nasution et al., 2020).

2.4.4 Nitrate Sensor

Nitrate (NO3), also referred to as a salt, is a nitrogen and oxygen combination. Nitrogen is released by decomposing matter such as plants, human waste, and animal waste. Nitrates are necessary for plant growth because they aid in the formation of amino acids and proteins. Nitrate is extensively water soluble, and excess nitrate that is not utilized by plants can seep into groundwater.

The content of nitrate in water is measured with a nitrate meter. Nitrate occurs naturally in water and is not toxic at low concentrations. However, nitrate is dangerous to aquatic ecosystems at high concentrations and, if found in drinking water, can be risky to human health (AquaRead, n.d.).

2.4.5 Phosphorus Sensor

Phosphorus (P) appears in natural water in the form of organic and inorganic phosphates (PO4). Phosphate (PO43) ions are required to generate useful energy from direct sunlight. It promotes cell growth and reproduction. Therefore, a small amount of phosphorus loss from soil may prevent the growth of freshwater weeds and algae. When phosphorus is fed to the lake in excess due to trivial human activities such as urban sewage, street runoff, and rural home septic tanks, it also impacts the environment as well. Phosphorus may travel with groundwater and combine with surface water. As a result, high phosphorus in groundwater may have an impact on surface water quality, which is a major problem. Phosphate levels in water are high, which reduces the amount of dissolved oxygen in lakes and rivers. This phenomenon has a negative impact on the lives of plants and animals within rivers and lakes. Phosphates are not toxic to humans or animals unless present in large quantities. Fish may suffer stomach problems if they live in water with a high level of phosphates. Phosphate levels in fisheries typically range from 0.005 ppm to 0.05 ppm. Periodic blooms are caused by phosphorus levels ranging from 0.08 ppm to 0.10 ppm. However, if phosphate levels continue at 0.05 ppm for an extended period of time, the eutrophication process will be halted. As a result, for economic and environmental reasons, it is critical to have excellent phosphate management in water used in farming, lakes, or rivers (Akhter et al., 2021). Thus, Phosphorus sensors are designed to monitor the phosphorus level of a given water.

2.4.6 Electrical Conductivity Sensor

An analog electrical conductivity meter tests the electrical conductivity of aqueous solutions and subsequently assesses the water quality. These sensors are commonly used in water culture, aquaculture, environmental detection, and other industries (DFROBOT, n.d.). Additionally, EC sensors are used to test and identify the quality of industrial process water, human drinking water, marine features, and battery electrolytes (Utmel, 2022).

2.5 Machine Learning Algorithm

Generally, costly and time-consuming laboratories and statistical analyses have been used to determine water quality. A more efficient and affordable approach is required due to the catastrophic effects of low water quality. The study investigates several supervised machine learning algorithm techniques to estimate the water quality class and the water quality index since the water quality class and the water quality index are one of the processes to characterize the general quality of water (Harnsoongnoen, 2021).

2.5.1 Random Forest Regression

The most efficient model was created using the Random Forest method, which produced the most trustworthy metrics when compared to the other techniques. The percentage errors of predicted and measured DO levels differ significantly when a predictive model created using the Random Forest Regression (RFR) and Decision Tree Regression (DTR) algorithms is used (Harrison et.al., 2021) Random forest (RF) is a non-linear regression model that can be used in time-series forecasting applications. The RF algorithm consists of an ensemble of decision tree models. The input data is recursively partitioned into two groups based on specified criteria in a decision tree until a predetermined stopping criterion is fulfilled (Insausti et al., 2020).

The study uses a decision tree-based random forest classification method. The idea behind a decision tree algorithm is to train a model by building a series of nodes that divide an observation into different classes based on predictors. The algorithm's objective is to divide the predictors in a way that minimizes tree correlation, lowering the procedure's overall variance and improving predictive accuracy. Adding more randomness to the tree-growing process lessens the correlation (Franco, 2019).

RF model to predict algal blooms in the freshwater Urayama Reservoir and the saltwater Lake Shinji, both in Japan. To predict chl-a content one to six months in advance, the researchers analyzed monthly water records with more than ten years' worth of data. The outcomes demonstrated that the RF model was able to predict the broad trends in chl-a concentration. Additionally, the model enabled the scientists to identify the factors that had the greatest influence on the forecast: pH, total nitrogen/total phosphorus, and chemical and biochemical oxygen demand (Kaidarova et al., 2018).

2.6 Microprocessor

A microprocessor is a device also referred to as a processor or central processing unit in charge of processing tasks and instructions. It is intended to carry out rational operations, interprocess communication, device communication, and input/output activities in computing management (Nag et al., 2017). A sensor that utilizes a microprocessor can increase the sensor's usage and track the parameters in a system (Ahmed et al., 2019).

2.6.1 Arduino Uno

Devices that can interface with the environment are built using Arduino's open-source platform. It can read inputs and translate them into outputs with the aid of its internal microcontroller (IEEE Computational Intelligence Society. Philippines Chapter, 2019). The Arduino Mega ATmega2560 is used in the study. It contains 16 analog inputs, 4 UARTs (hardware serial ports), a 16MHz crystal oscillator, 54 digital I/O pins (15 of which can be utilized as PWM outputs), a USB connector, a power jack, an ICSP header, and a reset button. It possesses every part needed to support a microcontroller. Simply connect it to a computer with a USB cable, or power it on with an AC-DC adapter or battery to start proceeding. Most shields designed for Uno and earlier Duemilanove or Diecimila boards are compatible with the Mega 2560 board (Cruz et al., 2021).

2.6.2 NodeMCU esp8266

Given that it is inexpensive and offers a versatile method of connecting objects to the internet, the NodeMCU ESP8266 is a development board that is frequently used for Internet of Things (IoT) applications. It supports Wi-Fi and programming, making it possible to quickly develop and implement IoT applications. The board's compatibility with the Arduino IDE and a number of libraries simplifies programming. Due to its compact shape and low power consumption, it is ideal for a variety of applications, including industrial control systems and home automation. NodeMCU, a platform for building open-source hardware, software, and firmware, was initially developed for the LUA-based ESP8266 Wi-Fi SoC device.

2.6.3 Pocket Wifi

A pocket Wi-Fi device refers to one that connects to the Internet over a wireless local area network (WLAN). A pocket Wi-Fi device serves as a hotspot, allowing cellphones, tablets, laptops, and other devices to establish a connection to the Internet. Travelers can also benefit from pocket Wi-Fi gadgets. Simply buy a local SIM card in your destination country and plug it into your pocket Wi-Fi to connect to the Internet at local rates while maintaining your primary phone number activated to receive calls (Beyeler, n.d).

2.7 Website Application

A website application, also referred to as a web application, is a software program that can be accessed through a web browser or a web-based interface. Its purpose is to offer users functionality and services via the Internet. Web applications are constructed using web technologies like HTML, CSS, and JavaScript and operate on web servers. Users can access these applications on different devices, including desktop computers, laptops, tablets, and smartphones (Contributor, 2023).

2.7.1 Hypertext Markup Language (HTML)

HTML is a widely used markup language utilized in the creation of web pages. Its purpose is to establish the structure and content of a webpage by utilizing a range of tags and elements. These tags and elements play a crucial role in defining various components of a webpage, including headings, paragraphs, images, links, and forms. When interpreted by web browsers, these tags and elements enable the structured display of content (Lutkevich, 2020).

2.7.2 Cascading Style Sheets (CSS)

CSS serves as a style sheet language specifically designed to describe the presentation and visual arrangement of documents written in HTML or XML. By employing CSS, web designers can implement a collection of rules and properties that determine the appearance of elements on a webpage. This allows for precise control over factors such as fonts, colors, spacing, layout, and other visual aspects, all of which can be altered independently of the actual content (O'Grady, 2023).

2.7.3 Javascript

JavaScript is an advanced programming language that empowers websites with interactive and dynamic capabilities. Its main purpose lies in client-side scripting, allowing web browsers to execute code that enhances the functionality and interactivity of web pages. JavaScript can manipulate and alter webpage content, respond to user actions, carry out calculations, validate forms, and establish communication with servers for data retrieval or transmission (Paruch, 2023).

2.7.4 PHP Programming Language

PHP is a widely utilized server-side scripting language in web development. Its purpose is to generate dynamic web pages and applications that facilitate interactions with databases and manage server-side operations. PHP code can be seamlessly integrated within HTML code, enabling developers to combine PHP logic with HTML markup. This code is executed on the server side, ensuring that the processed output is transmitted to the client's web browser (Toal, 2022).

2.7.5 mySQL Database

According to Richard (2023), for storing and managing structured data, a lot of people utilize the open-source relational database management system (RDBMS) known as MySQL. It offers a reliable and expandable framework for quickly classifying, accessing, and modifying massive amounts of data. MySQL follows the relational model, utilizing tables with rows and columns to store data in a structured manner. Users may perform a variety of activities, including querying, inserting, updating, and removing data, due to the support for SQL (Structured Query Language), a standard language for managing databases. MySQL is known for its stability, performance, and extensive community support.

2.7.6 Python Programming Language

Python is an interpreted, high-level, general-purpose programming language. Python's emphasis on readability lowers the cost of program maintenance because its syntax is straightforward and simple to learn (Brignell &

Dorey, 1983). This study has shown that python is the most viable programming language used to implement the machine learning algorithms, Random Forest regression, and the user interface (UI) of the web application developed in this study. The researchers aimed to fully understand a program's internal workings and deeper concepts while building the system with python.

2.8 Cloud Computing

According to the NIST (2018), cloud computing is the internet-based sharing of computing resources including storage, processing power, and software programs. These resources are available to users whenever they need them without the need for local infrastructure or hardware. Users can employ scalable computing resources, execute applications according to their needs, and store and access data remotely.

2.8.1 Internet of Things (IoT)

The internet of things (IoT) is a network of physical objects that connects and shares data with other online systems and devices through the use of sensors, software, and other technologies (Oracle Philippines, n.d.). According to Ashton, the Internet of Things (IoT) unifies the connections between human culture's "things" and our digital information system, or "the Internet.” (Truong, 2021). The study used this to connect the electronic device and the upcoming web application.

2.9 Buoy

Buoys are typically used to mark off maritime areas, record oceanographic, meteorological, and acoustic data, or mark the location of sensors or structures positioned at the bottom of the sea and protect them from physical harm (Agarwal, 2019).

Chapter 3

METHODOLOGY

This chapter discussed the research methodology that was used in the study: research design, procedure, data collection method, the sources of data, researcher's instrument construction and validation of instruments, distribution, and retrieval of instruments.

3.1 Research Design

3.1.1 Developmental Research

The research design that the proponents used in this study is the Developmental Research since it is focused on the development and innovation of the device and technology which is in line with the goal of the study. A quantitative method was implemented in the study as it involved collecting and measuring numerical data and seeking answers to quantitative questions.

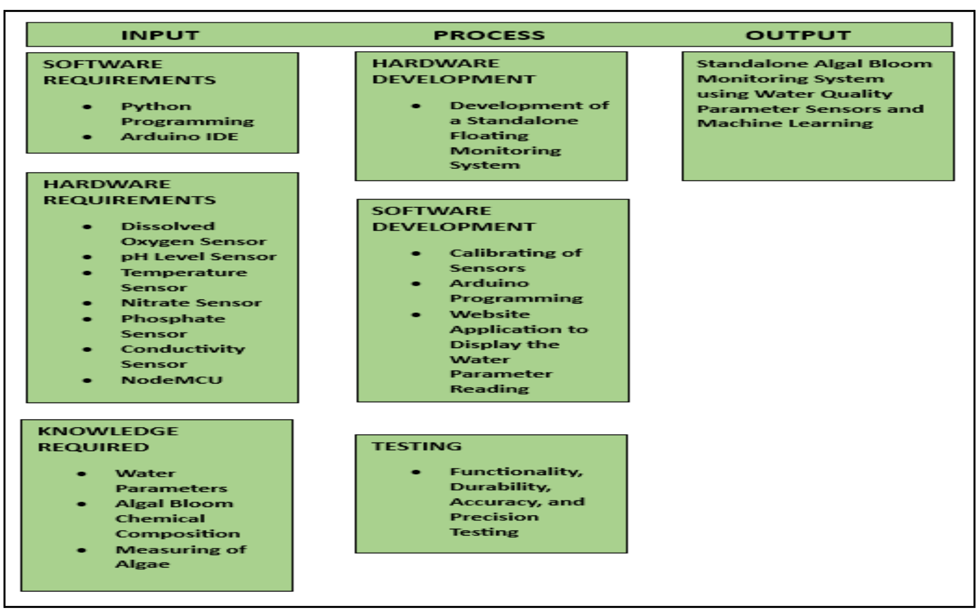


Figure 3.1 IPO *Model*

Figure 3.1 shows the IPO model of the study on which the input contains all the required instruments to be used in building the device. Software requirements are the programs needed to create a web application interface of the device where the data collected by the device would be shown. Similarly, the hardware requirements are for the creation of Standalone Algal Bloom Monitoring devices. In line with that, water quality parameter sensors are connected to Arduino Uno. Software development is necessary to accurately measure and gather data to predict algal bloom occurrence.

3.1.2 Data Acquisition

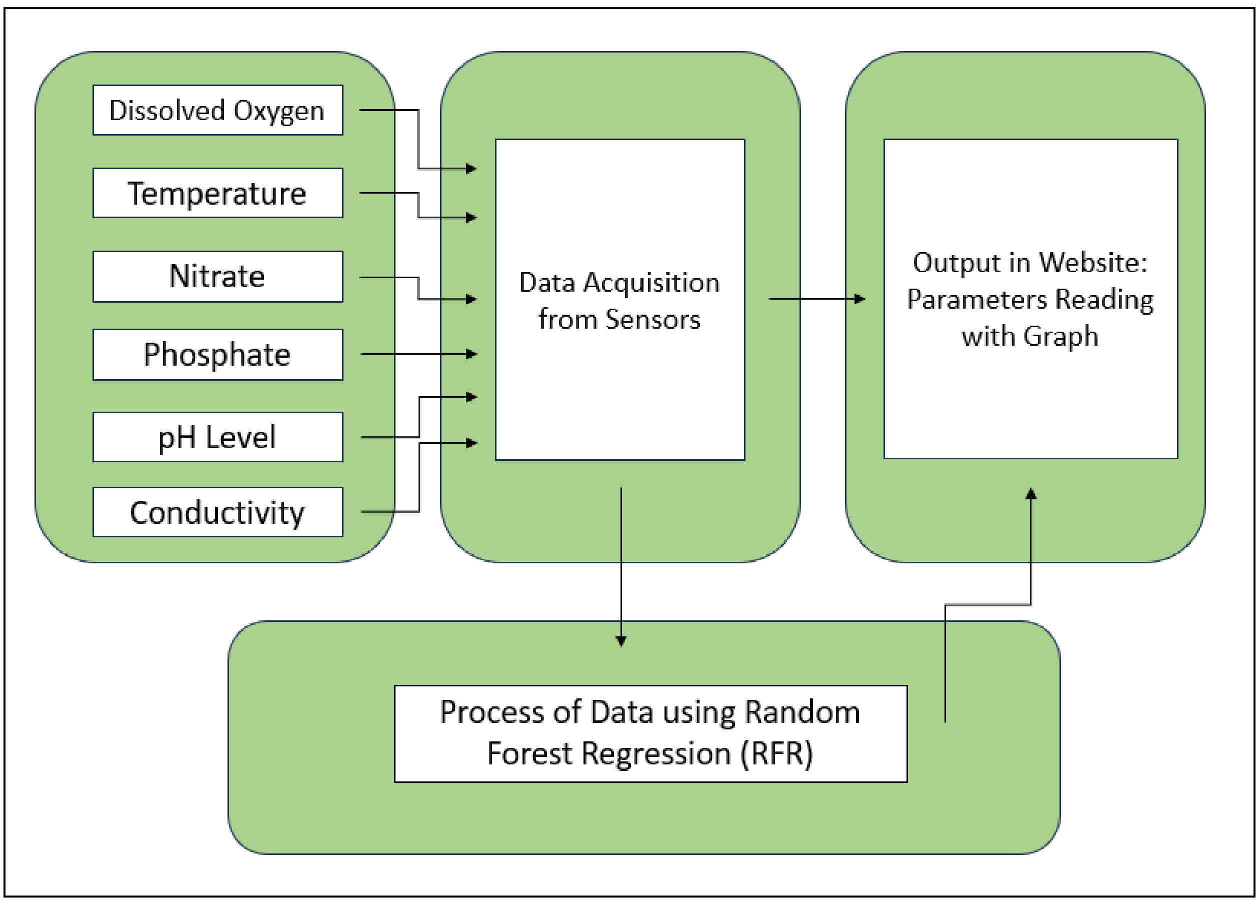


Figure 3.2 *Data Acquisition IPO*

3.2 Research Flow

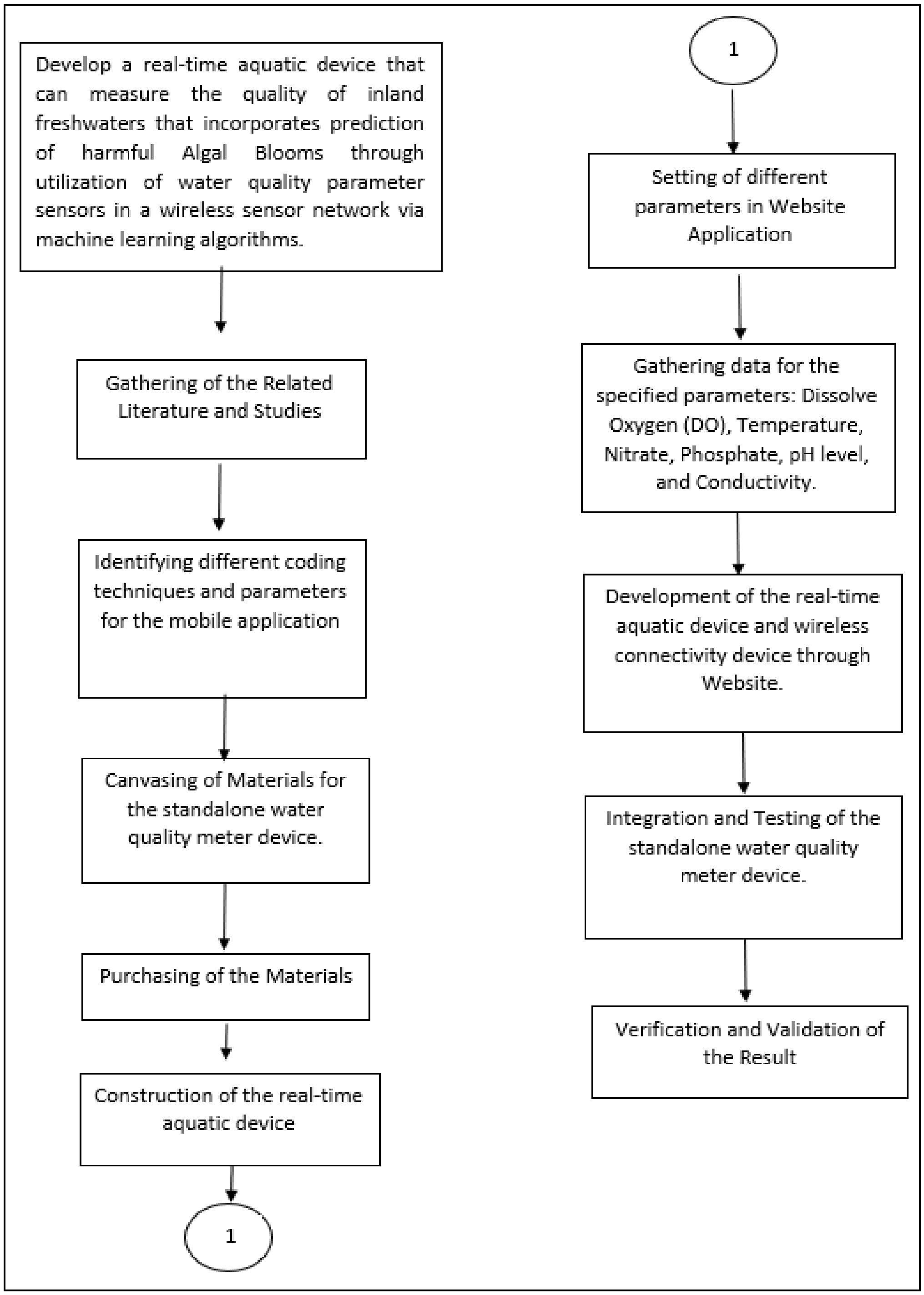


Figure 3.3 *Research Flow*

Figure 3.3 demonstrates how the research process flow actually occurred in sequential sequence throughout the study. The process started with conceptualizing how the real-time monitoring aquatic device came to be the main area of the concentration of the study. The researchers examined studies that had already been completed in order to determine what needed to be altered and modified. Upon constructing the idea for the device, the researchers started setting different parameters in the web application. Lastly, after the testing and integration of the real-time monitoring aquatic device, the validation and verification of the results.

3.3 Water Monitoring Buoy with Sensors

This section presented the different methodologies in the development of the hardware prototype, concepts, and parameters for the standalone algal bloom prediction system utilizing water quality sensors. The hardware requirements of the prototype are discussed in this section.

Specifically, the standalone algal bloom monitoring system that was developed is a water monitoring type of buoy since it utilized water quality parameters to predict the harmful algal bloom. A water monitoring buoy, in essence, measures different aspects of water, including water quality, currents, and waves. Among the parts of the aforementioned device are a data recorder, a buoy platform, telemetry gear, solar power, temperature string, mooring hardware, sensors, and sondes. The ideal buoy for a certain application could be made by adjusting the instrumentation and features of a water monitoring buoy station.

3.3.1 Design of the Standalone Algal Bloom Monitoring Buoy

The buoy is designed in consideration of the specifications observed and consolidated with design considerations from the experts. The buoy is quadratically shaped to have stable buoyancy in water. The buoy comes with a considerably heavy load at the bottom of it for stability purposes as well.

To ensure the safety of the sensors that were submerged in water, it had a plastic enclosure with tiny holes on its side. Fish typically attack prey smaller than them and avoid prey significantly larger than them, according to the University of Wisconsin Sea Grant Institute. Hence, as the fish of the beneficiary's pond or a specific body of water are typically Tilapia, Bangus, and other smaller fish, the boy's size is designed to be bigger than those fishes. Figure 4 to 6 shows the different views of the designed buoy.

Figure 3.4 *Cynosens Device*

3.3.2 Dimension of the Buoy

The buoy is composed of a solar panel with the dimension of 9 ⅛ " x 1'1 ⅞ ", and the container which protects the sensors with the dimension of 9 " x 6 " . The dimension of the top view of the buoy is 2 '2" x 2 '2". The side view dimension is 1'5 3/16" x 11".

3.3.3 Research Procedure

3.3.3.1 Materials and Equipment

Table 3.1 Materials and Equipment

Sensor	Microcontroller	Power System	Peripherals
Dissolved Oxygen Level Sensor	Arduino Uno	Powerbank	Connecting Wires
Temperature Sensor	NodeMCU	Solar Panel	Device Chassis and Insulation
pH level Sensor			
Nitrate and Phosphorus Sensor (NPK)			
Electrical Conductivity Sensor			

3.3.3.2 Sensors

This section presents and discusses the different sensors that were used in this study.

3.3.3.2.1 Dissolved Oxygen Sensor

Dissolved oxygen (DO) sensors can be designed for biochemical oxygen demand (BOD) testing, spot sampling, or long-term monitoring applications. A data logging system, water quality sonde, or dissolved oxygen meter can record measurement information obtained with a DO sensor. A DO sensor measures the amount of oxygen that has been dissolved directly into the water. Dissolved oxygen is crucial for the survival of fish and other aquatic creatures, making it one of the most significant indicators of water quality. Fish, plants, and other aquatic species cannot survive in water that has too little dissolved oxygen. Figure 3.5 shows the image of the dissolved oxygen sensor that will be used.

Figure 3.5 *Dissolved Oxygen Sensor*

3.3.3.2.2 pH Level Sensor

A pH Level sensor is one of the most significant devices frequently used to test water. This type of sensor can determine the acidity and alkalinity of water and other fluids. When used effectively, pH sensors can ensure the production processes in a manufacturing or wastewater facility, as well as the safety and quality of a product. The SEN0169 analog pH sensor is designed specifically to determine a solution's acidity or alkalinity by determining its pH. Aquaponics, aquaculture, and environmental water testing are just a few of the uses for this sensor. Figure 3.6 shows the image of the pH level sensor that was used.

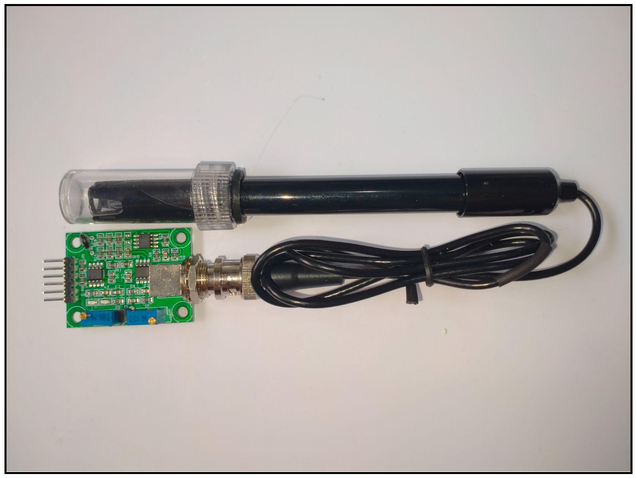


Figure 3.6 pH *Level Sensor*

3.3.3.2.3 Temperature Sensor

A temperature sensor is used to measure the temperature of water. Maxim Integrated makes the DS18B20, a 1-wire programmable temperature sensor. To monitor temperature in adverse circumstances, it is commonly utilized. The DS18B20 sensor can measure a wide range of temperatures, from -55° to +125°, with a reasonable accuracy of 5°C. Because each sensor has a unique address and only takes one MCU pin to relay data, it is a very suitable solution for measuring temperature at numerous locations without using up a lot of your digital pins. The image of the temperature sensor that was employed is depicted in Figure 3.7.

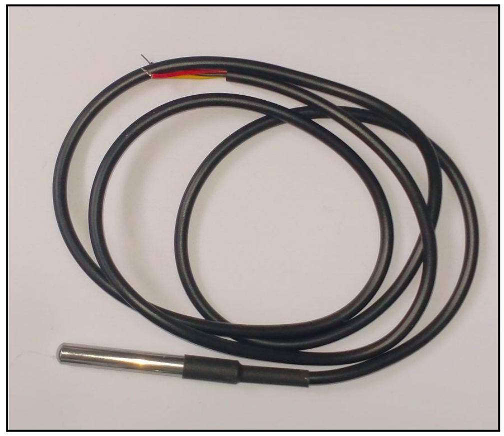


Figure 3.7 *Temperature Sensor*

3.3.3.2.4 Nitrate and Phosphorus Sensor

NPK sensor measures the nitrogen, phosphorus, and potassium level and is also used in many different applications such as aquaponics projects and systems. NPK Sensor and Arduino may be used to easily measure the amount of nutrients. To establish the level of nutrient present in soil or water, the content of N (nitrogen), P (phosphorus), and K (potassium) must be measured. This sensor assisted the user in determining nutrient shortfall or abundance.

Figure 3.8 *NPK Sensor*

3.3.3.2.5 Electrical Conductivity Sensor

The DFRobot Electrical Conductivity Sensor, in particular, covers a 3 to 5 volt input range and is compatible with both 3.3 and 5 volt main control boards. The output signal has low jitter and is hardware filtered. An alternating current signal is used as the excitation source, which improves precision, prolongs probe life, and effectively removes the polarization effect. The software library uses a two-point calibration mechanism and has the ability to automatically identify the standard buffer solution, giving the user a straightforward and practical sensor. With this sensor, a primary control board (such an Arduino), and the software library, it is possible to quickly construct an electrical conductivity meter and start plugging it in. In addition to meeting the requirements of various water quality tests, DFRobot provides a variety of water quality sensor devices with consistent sizes and interfaces that are suitable for home-made multi-parameter water quality testers. Conductivity, which relates to a material's capacity to carry electricity, is the inverse of an object's resistivity. The conductivity of a liquid solution is a measure of its capacity to conduct electricity. Conductivity is an important water quality metric. It can reflect the concentration of electrolytes in water (DFROBOT, n.d.).

Figure 3.9 *Analog Electrical Conductivity Sensor*

3.3.3.3 Microcontroller

3.3.3.3.1 Arduino UNO

The Arduino UNO is an open-source programmable microcontroller board that is inexpensive, versatile, and user-friendly and can be used in a wide range of electronic applications. This board can communicate with other Arduino boards, Arduino shields, and Raspberry Pi boards as well as control relays, LEDs, servos, and motors as output devices.

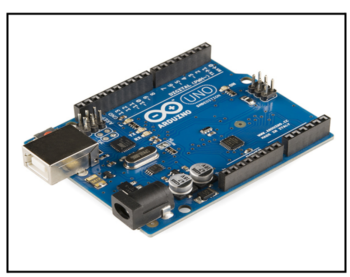


Figure 3.10 *Arduino UNO*

3.3.3.3.2 NodeMCU esp8266

The ESP8266, a low-cost System-on-a-Chip (SoC), is the foundation of the NodeMCU (Node MicroController Unit), an open-source software and hardware development environment. All basic computer parts, including CPU, RAM, networking (WiFi), and even a modern operating system and SDK, are included in the Espressif Systems ESP8266. It is therefore an excellent option for all kinds of Internet of Things (IoT) projects.

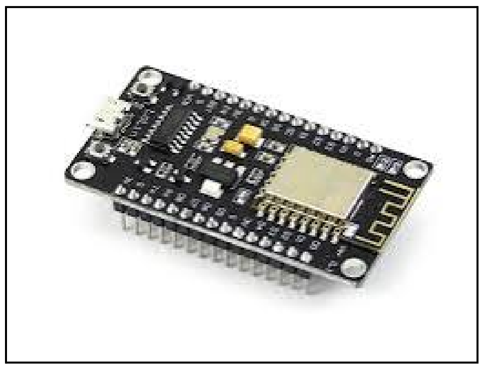


Figure 3.11 *NodeMCU esp8266*

3.3.3.4 Power System

3.3.3.4.1 Powerbank

A power bank is a portable charger that enables on-the-go recharging of electrical gadgets. Small, portable devices to larger, higher-capacity Power Banks are available in a variety of sizes. In a similar way, powerbanks are utilized to recharge tablets, GoPros, portable speakers, GPS units, smartphones, cameras, and MP3 players. A Power Bank can be used to charge every device that uses USB power. Only regular charging is required for it as well. A solar panel is then attached to the powerbank so that it may recharge on its own. The image of the powerbank that was used is depicted in Figure 3.12.

Figure 3.12 *Romoss Power Bank*

3.3.3.4.2 Solar Panel

A solar panel is one aspect of a photovoltaic system that transforms sunlight, composed of energy particles known as "photons," into electricity. They are constructed from a collection of solar cells organized in a panel. They work together to generate electricity and come in a range of rectangular shapes. The power bank's USB port would be used to connect the solar panel. The image of the solar panel that was employed is depicted in Figure 3.13.

Figure 3.13 *CL-1615 16V 15W Solar Panel*

3.3.3.5 Peripherals

3.3.3.5.1 Connecting Wires

Jumper wires will be used in connecting circuit boards, components, and other modules to its designated port of connection within the circuit. Figure 3.14 shows the image of the jumper wires that were used.

Figure 3.14 *Jumper Wires*

3.3.3.5.2 Device Chassis and Insulation

Plywood was used as the main material for the chassis of the device. Specifically, it is the main frame of the buoy. Plywood is a lightweight and moisture resistant board. It can be cut, routed, machined, heat-formed or bonded. Plywood features High Strength and Dimensional Stability, Water and Chemical Resistance. Because of its helpful features such as moisture resistance and high strength, plywood has become one of the most preferred building materials. Despite its strength and versatility, plywood is still an economical and long-lasting material for small businesses.

Figure 3.15 *Plywood*

Water-based acrylic-latex paints are often the easiest to work with on plywood projects. Yellow paint was being used to the buoy to easily recognize it when afar in the water.

Figure 3.16 *Paint*

Wall putty is a premium quality white cement-based putty with unique properties and silicone additives and waterproofing properties. Wall putty and epoxy were used to strengthen the chassis of the buoy and make it more water-resistant. Figure 3.17 shows the image of the Wall Putty and epoxy.

Figure 3.17 *Wall Putty (left) and epoxy (right)*

Foam was used to cover the inner surface of the chassis to lessen the heat absorbed by the device once it was deployed in a fishpond**.** Insulation foam is an excellent material used as thermal insulation for roofs, walls, etc. It is made from polyethylene, lightweight and flexible, making it easy to install on roofs and walls. Its closed cell feature makes it water and moisture resistant. Additionally, insulation foam is non fibrous and non-dusting. Also, it is made with fire retardant additives with a self-extinguishing feature. Figure 3.18 shows the image of the insulation foam that was used.

Figure 3.18 *Insulation Foam*

Floating device is salient to keep the buoy floating in the water. It was intended to give more buoyancy and help the buoy stay afloat in water.

Figure 3.19 *Floating Device*

3.4 Development of a program to predict the harmful algal bloom using Random Forest Regression

The researchers would predict the occurrence of algal bloom by monitoring the Dissolved Oxygen, Nitrate, pH, Water Temperature, Conductivity, and Phosphate levels present in the water. When it comes to the degree of importance of the parameters, Dissolved Oxygen, Nitrate, and Phosphate are the parameters mostly associated when Algal Bloom is present in water. Nitrate and Phosphate are the two main variables that affect eutrophication which leads to Algal Bloom while Dissolved Oxygen is consumed by bacteria when Algal Bloom is in decomposition state. Random Forest Regression algorithm was utilized because of its versatility, and it is the most reliable method for prediction.

3.4.1 Calibration of Water Quality Parameter Sensors

Various options for sensor calibration are considered to minimize the cost. The researchers first opt to go to different testing laboratories such as BFAR and LLDA that conduct water quality testing to calibrate the sensors that will be used. If there are no available laboratories for testing, purchasing different sensors would be the other option for the basis of sensor calibration.

3.4.2 Data Gathering for Training and Testing Sample

Machine learning models must digest large amounts of structured training data in order to build intelligent applications capable of understanding. The first step in solving any AI-based machine learning problem is to collect enough training data.

Various data sets were gathered and split into the training and testing set in Random Forest Regression.

3.5 Development of a website application to display the acquired data

Python is a high-level, interpreted, general-purpose programming language used for software development, data analysis and visualization**.** Python Programming was utilized to develop the web application for the device. The website application displays the different parameter readings. Wireless connection between the device and the web application was established using ESP8266 Wifi Module from Arduino Uno via Internet of Things.

3.6 Device Functionality Testing

The researchers tested the device's accuracy by conducting various tests in a controlled environment. The results of the tests were compared to the measurements gathered from existing Algal Bloom monitoring devices or laboratory testing. The device was calibrated until the lowest margin of error was reached for the comparison of the results.

3.6.1 Research Locale

In this study, the researchers envisioned deploying the project in a small-scale aquaculture industry. This project gave a huge help to fisherfolks in monitoring the parameters of their fishponds and alike and knowing ahead of time when the occurrence of harmful algal bloom. With the use of this device, aquaculture farmers can take precautionary actions to protect their livelihood.

The researchers coordinated with the Bureau of Fisheries and Aquatic Resources, Region IV-A in Los Banos, Laguna, to secure a location where the device was safely and effectively deployed for initial stages and training. Additionally, the project has been deployed in a fishpond located at Laguna Lake to further test the accuracy. This collaboration with them was intended to strengthen and solidify the study's conceptual foundations regarding pertinent procedures and relationships critical to the study.

3.6.2 Testing of Device

3.6.2.1 Accuracy Testing

The accuracy of the proposed device was tested by comparing the gathered data to the readings of EXO YSI Water Monitoring Device of BFAR. The proponents calibrated the proposed device to meet the standards of the device from BFAR.

3.6.3 Testing Procedure

For data acquisition, the researchers determined the relationship of each defined parameter to the level of water quality at which the harmful algal bloom occurred. Then, analyzed the values, specifically the last state of water quality before the algae have occurred. This was done by taking the correlation coefficient of those water parameters.

Based on the obtained results, the researchers found suitable input patterns. Consideration of the optimal combination of inputs was necessary to produce a model that accurately predicted harmful algal blooms.

3.6.4 Evaluation Procedure

According to published literature on hydrological model calibration, validation, and application, several techniques are recommended for model performance evaluation of hydrological time series forecasting.

The random forest regression model was being evaluated using statistical analysis. Water parameter sensors were being tested, and the gathered data was compared to the industrial monitoring equipment used by the Bureau of Fisheries and Aquatic Resources in Batangas. The floatation device gave enough buoyancy to the circuit compartment, and the support rope from four sides kept them together.

Table 3.2 Evaluation Procedure

Parameters	EXO YSI Water Monitoring Device of BFAR	Proposed Device
Dissolved Oxygen		

(mg/L)		
Temperature (°C)		
pH Level		
Nitrate (ppm)		
Phosphate (ppm)		
Conductivity (ms/cm)		

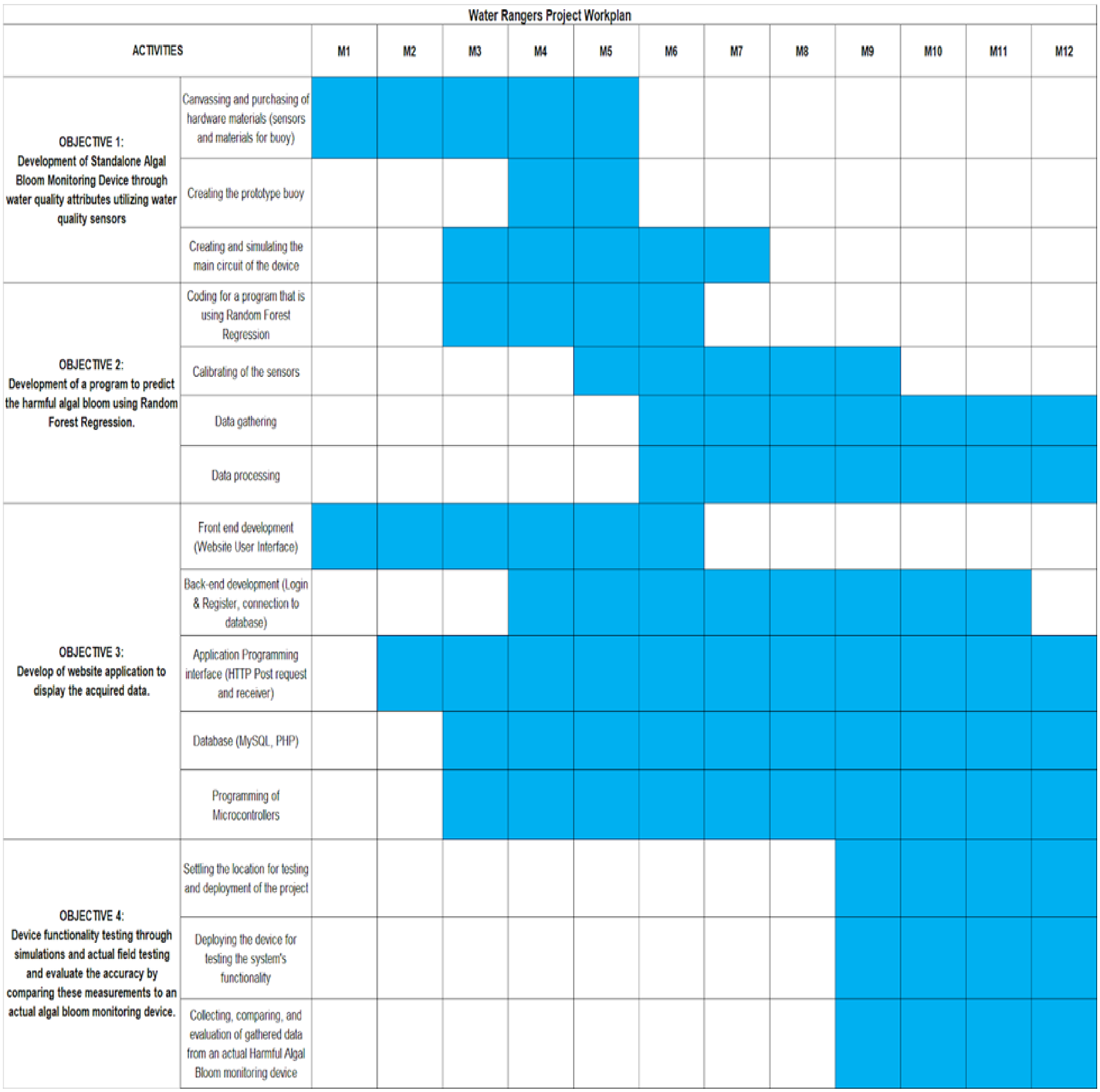
3.7 User-Acceptance and Field Testing of Algal Bloom Monitoring System

The testing and evaluation of the device functionality occurred at Batangas Inland Fisheries Technology Outreach Station and Laguna Lake, located southwest of Luzon. The testing consisted of both simulations and actual field testing. The researchers coordinated with the Laguna Lake Development Authority and Bureau of Fisheries and Aquatic Resources Region IV-A (BFAR) in order to acquire authorization for field testing and calibration.

3.8 Statistical Analysis

In this study, the following statistical measures were considered by the researchers in this study: Coefficient of Determination (R2), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The Coefficient of Determination is a numerical measure which determines how well a model explains and predicts future outcomes. The Mean Squared Error measures how concentrated the data is around the best-fit line. The Root Mean Squared Error is a measure of the difference between the model's predicted and measured values. These measures were used to evaluate and analyze the machine learning model for the prediction of the device.

3.9 Project Work Plan (Gantt Chart)



Chapter 4

RESULTS AND DISCUSSION

This chapter discussed the results gathered from the testing of the preliminary device.

4.1 Project Technical and Description

This project is entitled CYNOSENS: IoT based Harmful Algal Bloom Prediction System using Random Forest Regression through Water Quality Parameters is a standalone monitoring buoy that sends the data gathered from six water quality sensors to website application through a wireless connection, whenever the system senses the fluctuation in the parameters and surpass the normal/standard level of the aforementioned parameters it then sends warning notifications to the website to alert the users about the level of the algae in their water system as a feedback mechanism. The buoy is a solar-powered device that automatically charges the battery embedded in the system.

The prediction system uses Random Forest Regression machine learning algorithms. Random Forest Regression is used in many studies to predict certain parameters as set to it based on the other features associated with the model. The sensors gather the water parameters data continuously and the MySQL database fetched the data from Arduino to save it and display it to the website parameter table for the user's end for monitoring.

In the processing stage, the Random Forest Regression processes the data received by the database to predict the level of the algae in the water and analyze whether it will be harmful to the water system. The algorithm is pre-trained to correlate the features of the six water parameters to the algal bloom level.

For the reflection and feedback stage, the reading of the sensors and warning of the algorithm was displayed and reflected through the website. The website shows the gathered data by the sensors with its chemical unit real time through the cloud base. The notification would appear at the lower right of the website, green colored text was shown if the state is normal otherwise red for bad water quality that could lead to rapid overgrowth of algae in the water. The web application can be used using desktop and mobile view.

4.2 Project Structural Organization

4.2.1 Parts of the System

The system is equipped with six water quality sensors namely: Temperature, pH, Nitrate, Phosphorus, Dissolved Oxygen, and Electrical Conductivity. It also has microcontrollers, 20000 mAh powerbank, sensors' module, rechargeable 12 V battery, and Nodemcu esp8266. The six water sensors gather the chemical properties of the water that are mainly involved in the occurrence of harmful algal bloom, it would process to be displayed on the be app.

The laptop unit comes with 512 Gb of SSD, with Windows 11 as an operating system. The computer is equipped with a 6 cores 12 threads 11th Gen Intel(R) Core (TM) i5-1135G7 processor. Running at a base clock speed of 2.42 GHz, 12 Gb of 3733 MHz ram at dual-channel configuration, and an Intel(R) Iris(R) Xe Graphics. The device served as the main processing unit for the Random Forest Regression algorithms.

The microcontroller used by the researchers for the project is Arduino Uno R3.

The Arduino Uno is responsible for controlling the sensors' analog and digital inputs and sending them to Nodemcu esp8266.

The solar panel is mounted on the buoy's cover, which converts sunlight into electrical energy to supply power to the battery. The main circuit is placed in the main body of the buoy, also called the circuit compartment, and it is tied to the floating device. The circuit compartment has insulation foam to protect the circuit from heat, and the outside body is covered by epoxy. The circuit compartment box has one hole that is securely perforated in the middle for the sensors to be able to access the water below the buoy. The monitoring process is handled by the buoy, or the hardware device and the web application is responsible for the feedback mechanism and algorithm process of the system. The images below depict the prototype hardware and software project to provide a visual representation.

Figure 4.1 *CynoSens Monitoring Buoy in Fishpond*

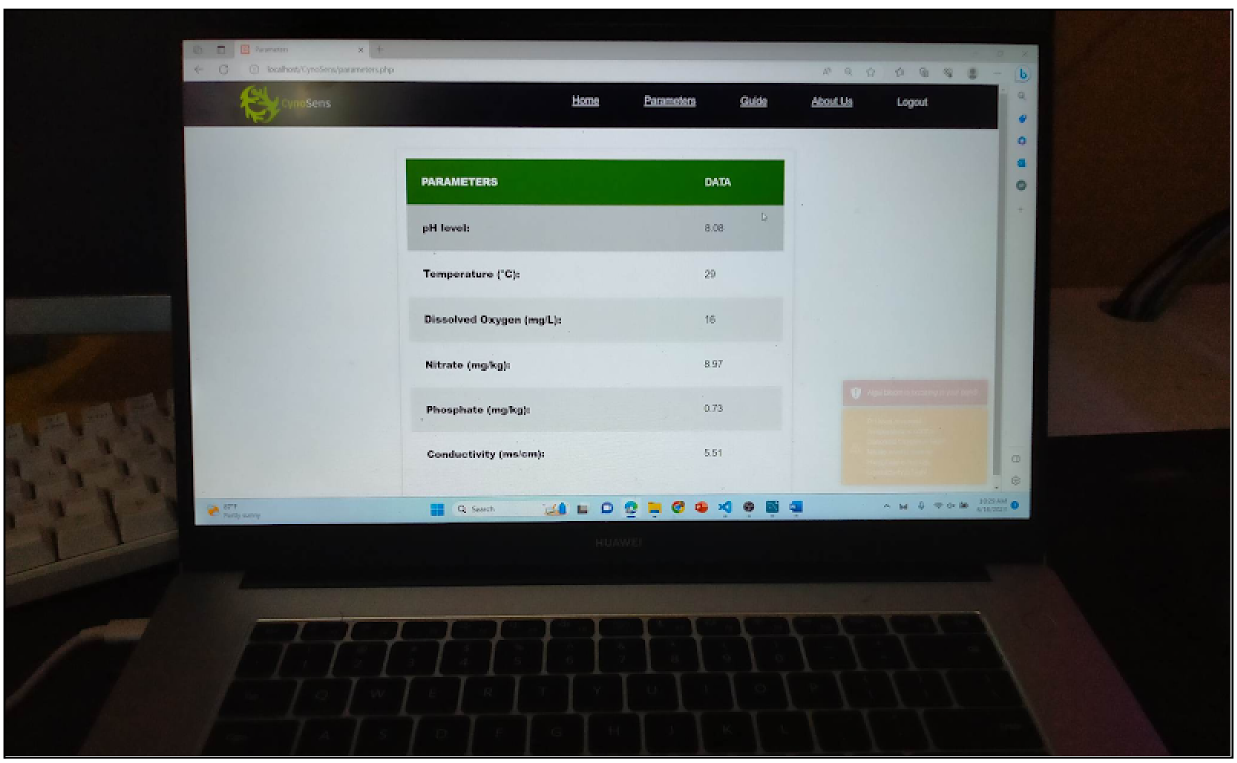


Figure 4.2 *Website view using Laptop*

4.3 Experimental Result and Data Analysis

4.3.1 Testing - Real Time Gathered Data

Table 4.1 *Real Time Sample of Gathered Data from Database - Data Gathered from Batangas*

id	phlevel	temperature	dissolvedoxygen	nitrate	phosphate	conductivity
1	7.82	27	8	8.52	0.64	3.48
2	7.45	27	9	8.05	0.96	3.80
3	7.40	29	10	8.26	0.59	3.96
4	7.23	27	10	7.42	0.50	4.36
5	7.29	27	10	8.40	0.54	4.78
6	7.97	27	8	7.42	0.73	4.02
7	7.15	27	10	8.19	0.82	4.99
8	7.56	29	9	8.96	0.80	4.01
9	7.82	29	10	8.04	0.53	3.05
10	7.76	27	9	7.64	0.97	4.56
11	7.19	27	10	8.07	0.85	4.98
12	7.27	29	8	8.71	0.51	4.28
13	7.83	28	10	8.04	0.62	4.57
14	7.65	29	8	8.62	0.86	4.88
15	7.12	29	10	7.74	1.00	4.83
16	7.46	29	8	8.79	0.57	3.77
17	7.50	28	8	8.35	0.88	3.11
18	7.01	29	9	8.59	0.88	4.52
19	7.45	27	10	8.06	0.62	4.37
20	7.69	28	10	7.65	0.95	4.28
21	7.65	28	8	8.30	0.94	3.42
22	7.31	28	8	8.32	0.98	4.49
23	7.27	28	9	7.04	0.75	3.10
24	7.75	27	9	8.85	0.69	4.84
25	7.64	27	9	7.16	0.80	3.49
id	phlevel	temperature	dissolvedoxygen	nitrate	phosphate	conductivity
25	7.64	27	9	7.16	0.80	3.49
26	7.26	28	10	8.33	0.65	4.92
27	7.26	28	9	7.22	0.69	3.55
28	7.75	28	10	7.91	0.71	3.51
29	7.59	29	9	7.14	0.57	3.01
30	7.60	29	9	7.84	0.60	3.86
31	7.38	27	9	7.08	0.67	3.51
32	7.57	27	10	8.91	0.58	3.19
33	7.94	28	9	8.91	0.75	3.56
34	7.11	29	8	7.15	0.88	4.69
35	7.72	28	8	8.96	0.70	4.23
36	7.79	28	10	8.45	0.94	4.82
37	7.24	28	9	7.06	0.56	4.49
38	7.35	27	8	8.70	1.00	4.22
39	7.02	27	10	8.78	0.86	3.87
40	7.74	28	10	8.03	0.75	3.34
41	7.90	29	10	8.76	0.84	3.42
42	7.32	28	8	7.58	0.76	4.68
43	7.99	29	8	7.83	0.89	3.79
44	7.31	28	9	7.98	0.97	3.28
45	7.98	28	8	8.43	0.82	4.24
46	7.45	29	10	8.62	0.67	3.47
47	7.21	27	10	8.65	0.72	3.63
48	7.50	27	9	8.97	0.98	4.39
49	7.00	29	8	8.09	0.65	3.58

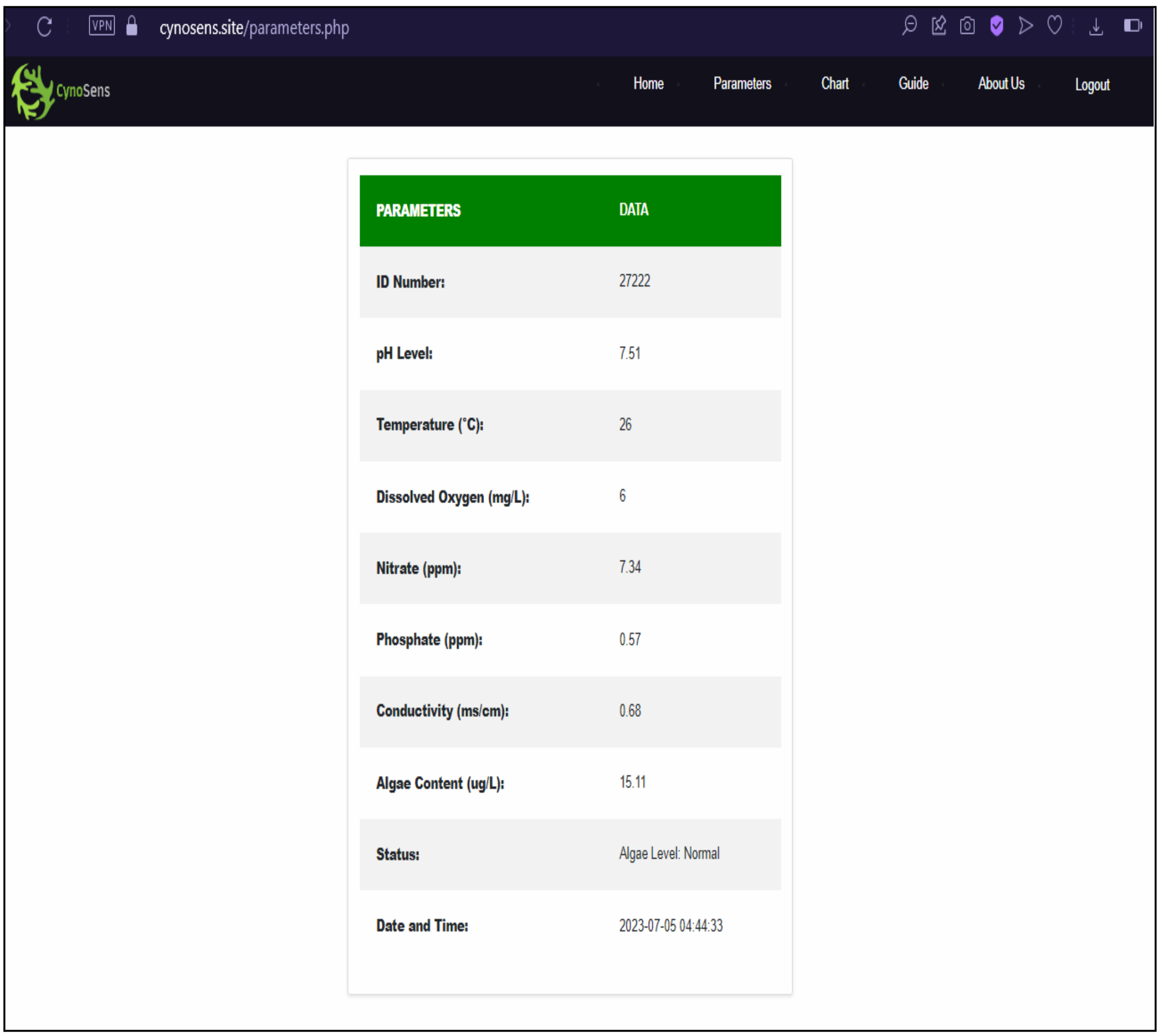
Table 4.3 shows the data collected at the Bureau of Fisheries and Aquatic Resources testing site in Tanauan, Batangas. The device was tested on a fishpond and left during the daytime for two days to monitor its water quality parameters and the algae content present in the water. As shown in the figure, the sensors gathered water parameters that are harmful to aquatic animals, especially the level of dissolved oxygen.

4.3.2 Deployment - Real Time Gathered Data

Table 4.2 *Real Time Sample of Gathered Data from Database - Data Gathered from Laguna Lake*

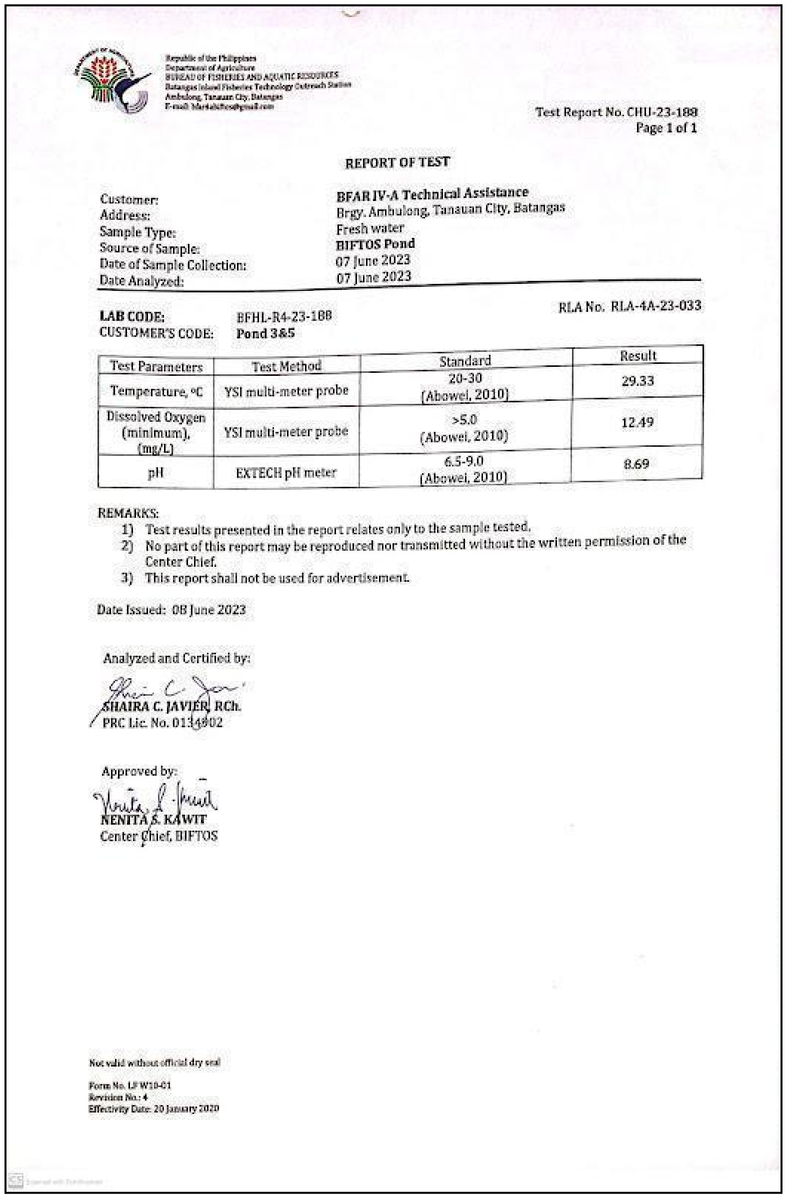
id	phlevel	temperature	dissolvedoxygen	nitrate	phosphate	conductivity	algaecontent	status	time
26901	7.61	26	6	7.39	0.57	0.69	15.34	Algae Level: Normal	2023-07-05 01:40:35
26902	7.44	26	6	7.32	0.56	0.66	15.30	Algae Level: Normal	2023-07-05 01:42:02
26903	7.51	26	6	7.43	0.54	0.66	15.75	Algae Level: Normal	2023-07-05 01:42:32
26904	7.46	26	6	7.32	0.59	0.69	15.89	Algae Level: Normal	2023-07-05 01:43:02
26905	7.69	26	6	7.59	0.55	0.70	15.49	Algae Level: Normal	2023-07-05 01:43:32
26906	7.64	26	6	7.39	0.56	0.68	15.75	Algae Level: Normal	2023-07-05 01:44:02
26907	7.40	26	6	7.34	0.54	0.70	15.42	Algae Level: Normal	2023-07-05 01:45:02
26908	7.21	26	7	7.39	0.58	0.66	15.55	Algae Level: Normal	2023-07-05 01:45:32
26909	7.40	26	7	7.47	0.57	0.67	15.41	Algae Level: Normal	2023-07-05 01:46:02
26910	7.44	26	7	7.34	0.54	0.70	15.79	Algae Level: Normal	2023-07-05 01:46:32
26911	7.42	26	7	7.47	0.54	0.70	15.32	Algae Level: Normal	2023-07-05 01:47:02
26912	7.64	26	6	7.56	0.59	0.69	15.61	Algae Level: Normal	2023-07-05 01:47:32
26913	7.60	26	7	7.31	0.58	0.70	15.76	Algae Level: Normal	2023-07-05 01:48:03
26914	7.64	26	7	7.56	0.54	0.69	15.84	Algae Level: Normal	2023-07-05 01:48:33
26915	7.45	26	7	7.58	0.57	0.71	15.34	Algae Level: Normal	2023-07-05 01:49:03
26916	7.52	26	6	7.44	0.55	0.70	15.40	Algae Level: Normal	2023-07-05 01:49:33
26917	7.21	26	7	7.35	0.59	0.66	15.89	Algae Level: Normal	2023-07-05 01:50:03
26918	7.60	26	7	7.30	0.59	0.69	15.70	Algae Level: Normal	2023-07-05 01:50:33
26919	7.43	26	7	7.57	0.54	0.69	15.54	Algae Level: Normal	2023-07-05 01:51:33
26920	7.20	26	7	7.30	0.56	0.68	15.30	Algae Level: Normal	2023-07-05 01:52:03
26921	7.39	26	7	7.32	0.57	0.69	15.80	Algae Level: Normal	2023-07-05 01:52:33
26922	7.53	26	6	7.49	0.54	0.68	15.76	Algae Level: Normal	2023-07-05 01:53:03
26923	7.70	26	7	7.36	0.59	0.70	15.81	Algae Level: Normal	2023-07-05 01:53:33
26924	7.42	26	7	7.34	0.56	0.66	15.40	Algae Level: Normal	2023-07-05 01:54:03
26925	7.48	26	6	7.49	0.54	0.70	15.40	Algae Level: Normal	2023-07-05 01:54:33
26926	7.42	26	6	7.51	0.55	0.70	15.37	Algae Level: Normal	2023-07-05 01:55:03
26927	7.30	26	7	7.34	0.55	0.70	15.74	Algae Level: Normal	2023-07-05 01:55:33
26928	7.21	26	7	7.40	0.56	0.66	15.74	Algae Level: Normal	2023-07-05 01:56:03
26929	7.60	26	6	7.41	0.54	0.66	15.61	Algae Level: Normal	2023-07-05 01:57:03
26930	7.29	26	7	7.50	0.55	0.67	15.50	Algae Level: Normal	2023-07-05 01:57:33

Table 4.4 shows the data collected at the deployment site in Laguna Lake. The device was deployed on a fishpond for several days to monitor its water quality parameters and the algae content present in the water. As shown in the figure, the algae content of the water is at the normal level of around 15 ug/L. The gathered data from six sensors are being shown as well.



PARAMETERS	DATA
ID Number:	27222
pH Level:	7.51
Temperature (°C):	26
Dissolved Oxygen (mg/L):	6
Nitrate (ppm):	7.34
Phosphate (ppm):	0.57
Conductivity (ms/cm):	0.68
Algae Content (ug/L):	15.11
Status:	Algae Level: Normal
Date and Time:	2023-07-05 04:44:33

Figure 4.3 *Sample of Gathered Data Reflected on Website*


Republic of the Philippines
Department of Agriculture
BUREAU OF FISHERIES AND AQUATIC RESOURCES
Batangas Inland Fisheries Technology Outreach Station
Ambulong, Tanauan City, Batangas
E-mail: [illegible]

Test Report No. CHU-23-188
Page 1 of 1

REPORT OF TEST

Customer: **BFAR IV-A Technical Assistance**
Address: Brgy. Ambulong, Tanauan City, Batangas
Sample Type: Fresh water
Source of Sample: **BIFTOS Pond**
Date of Sample Collection: 07 June 2023
Date Analyzed: 07 June 2023

LAB CODE: BFHL-R4-23-188 RLA No. RLA-4A-23-033
CUSTOMER'S CODE: **Pond 3&5**

Test Parameters	Test Method	Standard	Result
Temperature, °C	YSI multi-meter probe	20-30 (Abowei, 2010)	29.33
Dissolved Oxygen (minimum), (mg/L)	YSI multi-meter probe	>5.0 (Abowei, 2010)	12.49
pH	EXTECH pH meter	6.5-9.0 (Abowei, 2010)	8.69

REMARKS:
1) Test results presented in the report relates only to the sample tested.
2) No part of this report may be reproduced nor transmitted without the written permission of the Center Chief.
3) This report shall not be used for advertisement.

Date Issued: 08 June 2023

Analyzed and Certified by:

SHAIRA C. JAVIER, RCh.
PRC Lic. No. 0134902

Approved by:

NENITA S. KAWIT
Center Chief, BIFTOS

Not valid without official dry seal

Form No. LF W10-01
Revision No.: 4
Effectivity Date: 20 January 2020

Figure 4.4 *Laboratory Test From BFAR*

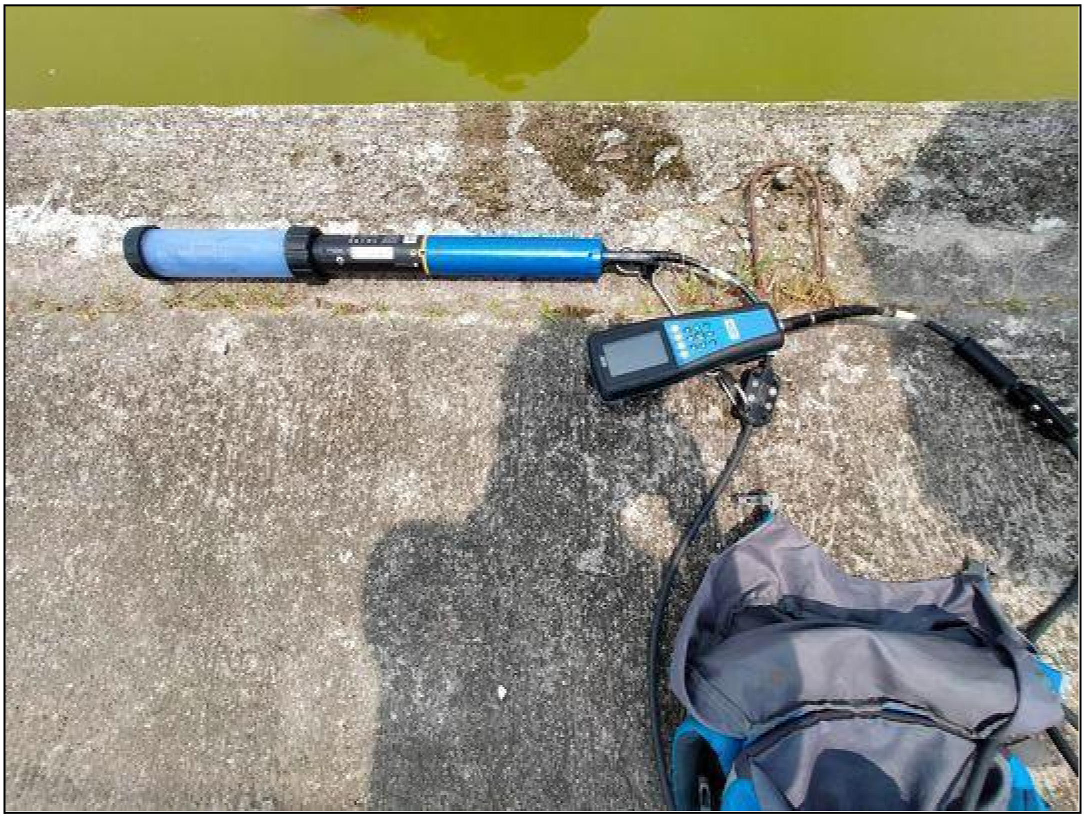


Figure 4.5 *EXO YSI Water Monitoring Device of BFAR*

Table 4.3 BFAR *Gathered Data by the EXO YSI device vs. CynoSens*

BFAR	Cynosens
Temperature: 29.33°C	Temperature: 30°C
Dissolved Oxygen: 12.49 mg/L	Dissolved Oxygen: 11.89 mg/L
pH: 8.69	pH: 8.5
Nitrate: 8.62 ppm	Nitrate: 8.97 ppm
Phosphate: 0.86 ppm	Phosphate: 0.70 ppm
	Conductivity: 4.88 us/cm

The table 4.8 above shows the gathered data by the device and the data resulted from the expert analysis. The device is tested at the Bureau of Fisheries and Aquatic Resources in Tanauan, Batangas. The monitoring device and the prediction buoy are placed in the water simultaneously to collect the water parameters. During the gathering of data, the researchers noted that they gathered data from their standard monitoring device to the prediction buoy system.

4.3.3 Evaluation of Random Forest Regression Prediction Model

In this study, Random Forest Regression machine learning algorithm was used to predict the alga content of the water using the other features which in this case are the water parameters. The created model is trained first using the dataset acquired and then hyper tuned using randomized search CV to enhance the predicted values it outputs. After hypertuning, the model was tested using the gathered values and evaluates its level of prediction.

The dataset has been split into training data (80%) and test data (20%). In the random forest regressor, the decision tree scales the input. The model is demonstrated by fetching the data of feature water parameters by the sensors in the database and predicting the Algae content using an HTML file. The file is put in a pickle file and the performances of the model are computed.

In [25]:
```
df_eval = pd.DataFrame(rf_random.predict(X_test), columns=['Prediction'])
y_test = y_test.reset_index(drop=True)
df_eval['Target'] = y_test

df_eval['Residual'] = df_eval['Target']-df_eval['Prediction']
df_eval['Differences%'] = np.absolute(df_eval['Residual']/df_eval['Target']*100)
df_eval
```

Out[25]:

	Prediction	Target	Residual	Differences%
0	9.5290	4.1	-5.4290	132.414634
1	1.2464	1.1	-0.1464	13.309091
2	5.6760	6.7	1.0240	15.283582
3	18.2490	23.9	5.6510	23.644351
4	0.5892	0.0	-0.5892	inf
...	...	...	...	...
125	3.7120	7.2	3.4880	48.444444
126	16.2820	10.5	-5.7820	55.066667
127	6.7000	8.0	1.3000	16.250000
128	11.8134	23.7	11.8866	50.154430
129	4.3710	1.5	-2.8710	191.400000

130 rows × 4 columns

Figure 4.6 *Sample result of the Testing Prediction of the Random Forest Regression Algorithm*

Figure 4.9 shows the sample result of prediction during the testing of the algorithm. The algorithm gathered the features then predicted the algae content value and compared it to the target or actual value.

In [19]:
```
df_eval.describe()
```

Out[19]:

	Prediction	Target	Residual	Differences%
count	130.000000	130.000000	130.000000	125.000000
mean	7.818417	7.934692	0.116275	inf
std	8.992351	17.384310	15.085688	NaN
min	0.000000	0.000000	-24.642000	0.726316
25%	2.372600	1.625000	-2.879250	19.249027
50%	4.984200	3.450000	-0.504650	48.806667
75%	9.742425	7.575000	0.727575	164.266667
max	57.222000	177.000000	160.214600	inf

Figure 4.7 Result *Evaluation of the Testing Prediction of Random Forest Regression Algorithm*

Figure 4.10 shows the model evaluation of sklearn. The performance measures are computed using the R squared Score, Root mean square error and Mean Square Error. The model's accuracy is determined using the built-in functions of Sklearn. If there is a larger than cutoff (MSE) gap between the test and the predicted value, the prediction was incorrect, if not, accurate prediction.

```
In [46]: from sklearn.metrics import r2_score

         print(f'Testing r2_score: {r2_score(y_train,rf_random.predict(X_train))}')

         Testing r2_score: 0.9148247255739896
```

Figure 4.8 Accuracy *of the Prediction Model using Random Forest Regression Algorithm*

The accuracy of the model calculated resulted in a score of 0.9148247255739806, hence the algorithm model accuracy is 91.48 %.

4.3.4 Website Application Interface

The web application displays the value of each parameter from the hardware sensors. The website can be accessed through any device because of its responsiveness which makes the display of the website fit the size of the screen of the device. This allows the website to be easily accessible to the users. The website contains: login and register, homepage, parameters, guide, and about us.

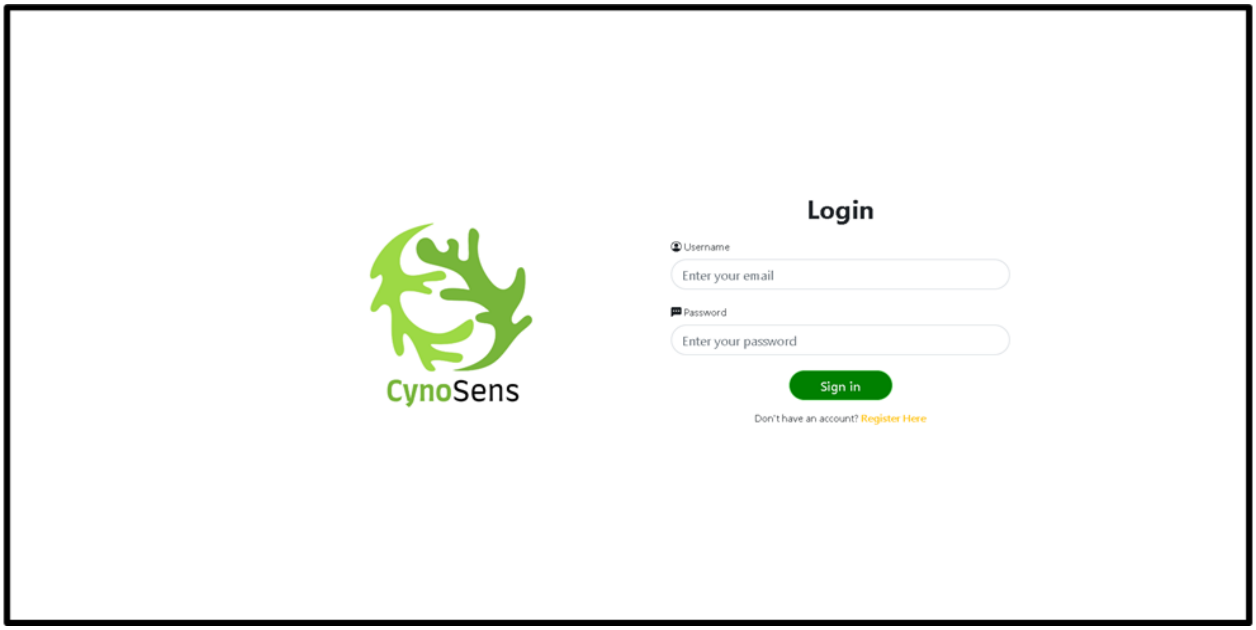


Figure 4.9 *Login page (Computer View)*

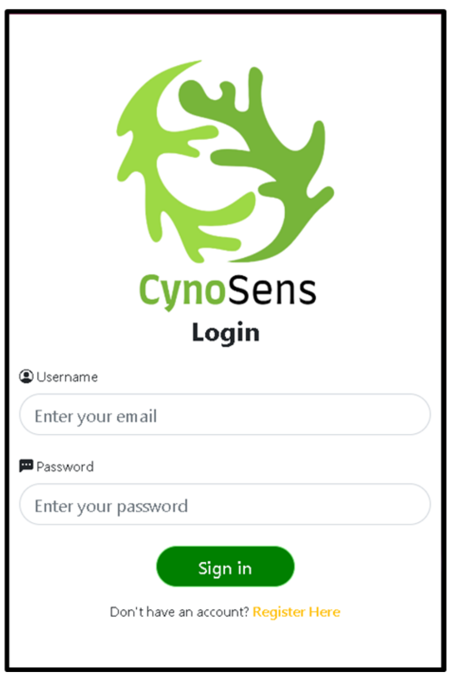


Figure 4.10 *Login page (Smartphone View)*

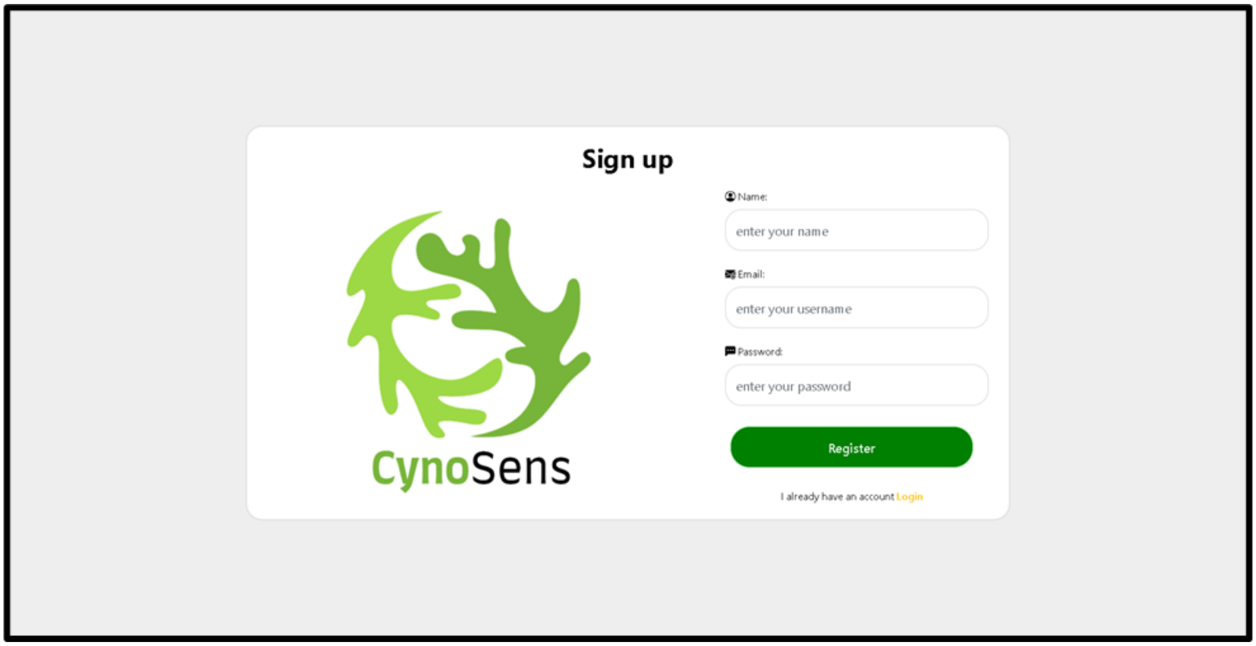


Figure 4.11 *Sign up page (Computer View)*

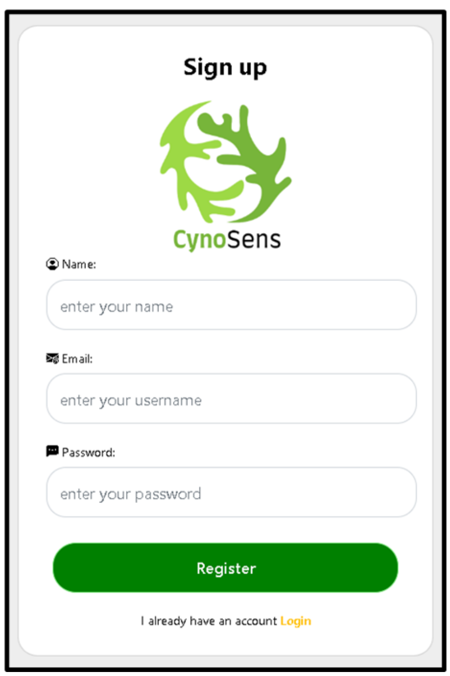


Figure 4.12 *Sign up page (Smartphone View)*

Figure 4.13 *Homepage (Computer View)*

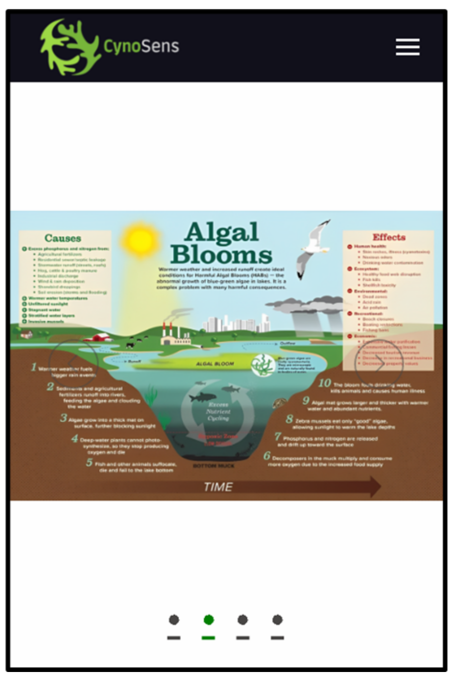


Figure 4.14 *Homepage (Smartphone View)*

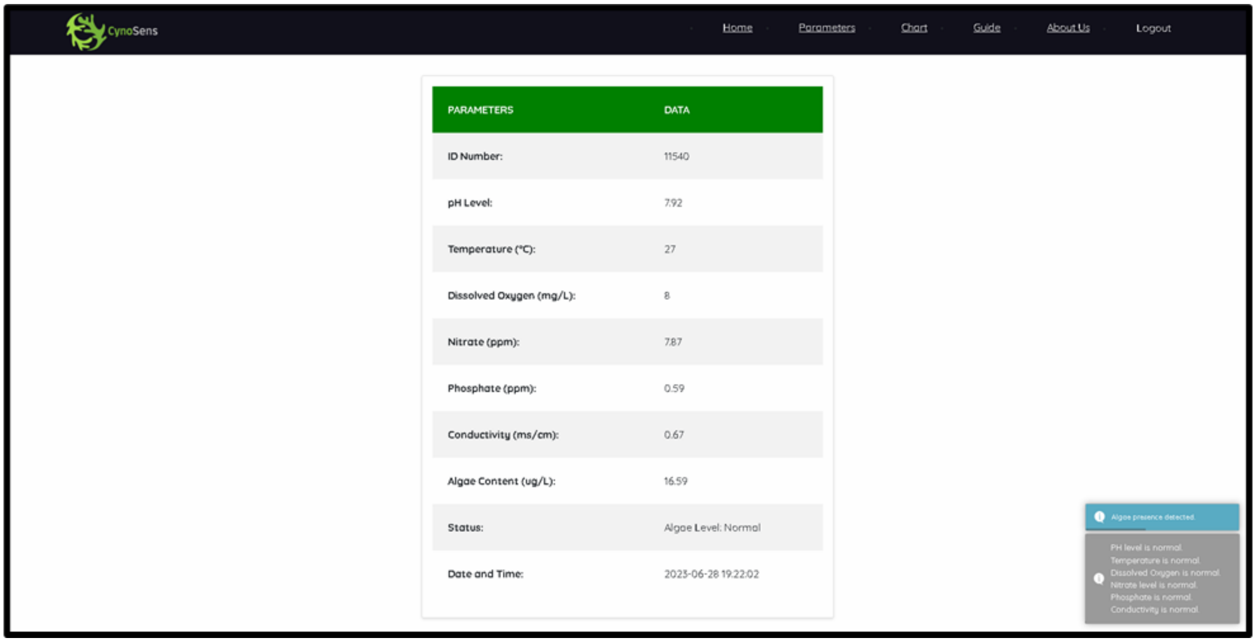


Figure 4.15 *Parameters (Computer View)*

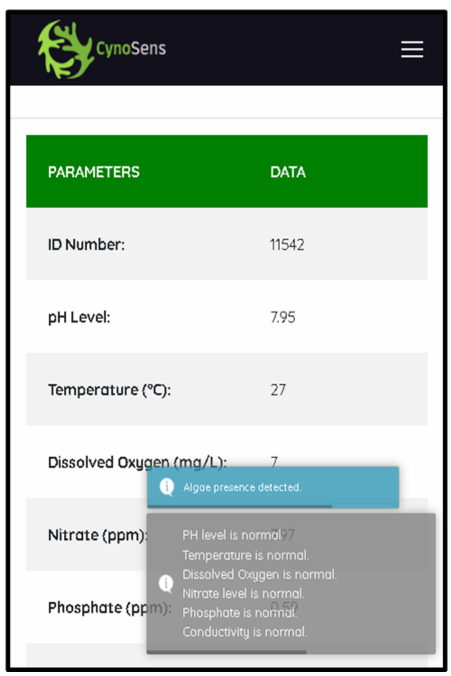


Figure 4.16 *Parameters (Smartphone View)*

Figures 4.15 and 4.16 showcases a designed parameters webpage that offers comprehensive insights. Within this webpage, a table is thoughtfully presented, featuring the six (6) water quality parameters alongside their corresponding IDs, dates, and times. Additionally, the table includes crucial information on algae level and content. One of the exceptional features of this webpage is its real-time data display, providing continuous updates every thirty (30) seconds. To ensure users are promptly informed, a synchronized notification pops up simultaneously with each data update. These notifications effectively communicate the parameter levels in terms of low, normal, and high categories, aligning with the standard water quality for type c pond waters from BFAR and the anticipated algae content.

Furthermore, an additional synchronized notification appears to present a prediction derived from an algorithm. This prediction serves to indicate whether an algal bloom is occurring, supplying valuable information regarding the algae level and content, along with an assessment of the water's safety for the pond's fish. The combination of real-time data updates, informative notifications, and predictive capabilities.

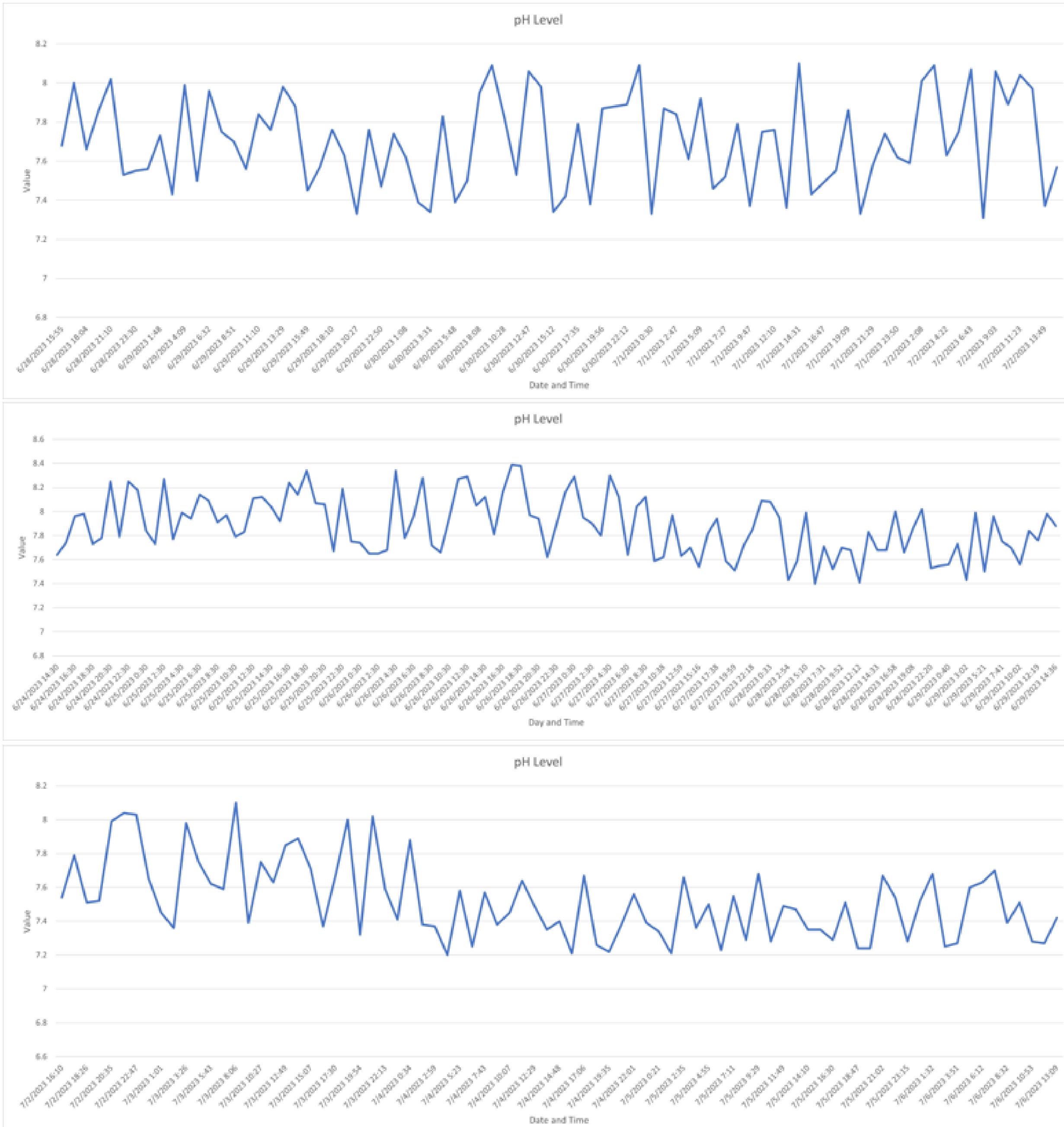


Figure 4.17 *Graphical Representation of Gathered Data for pH Level: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

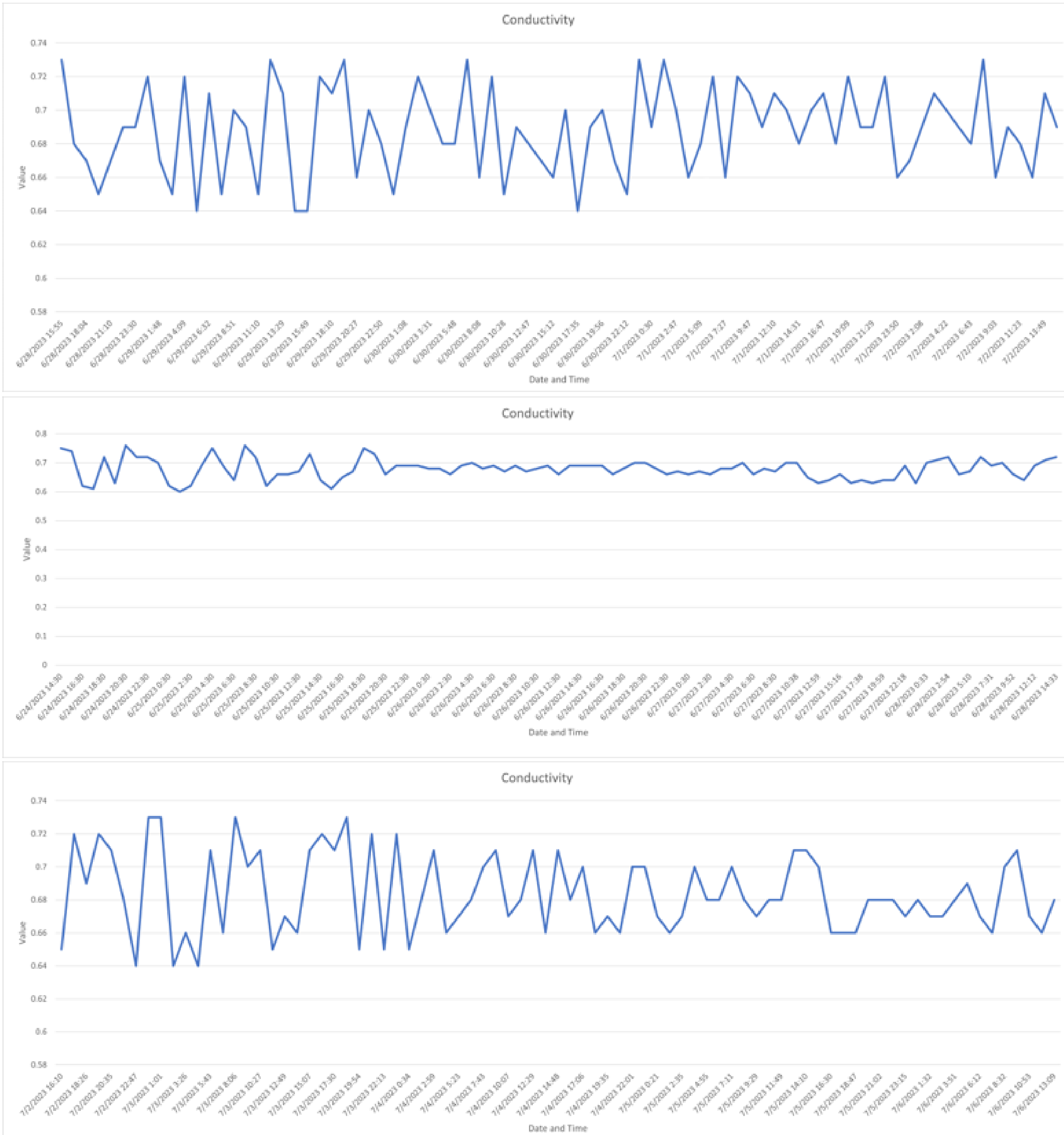


Figure 4.18 *Graphical Representation of Gathered Data for Electrical Conductivity: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

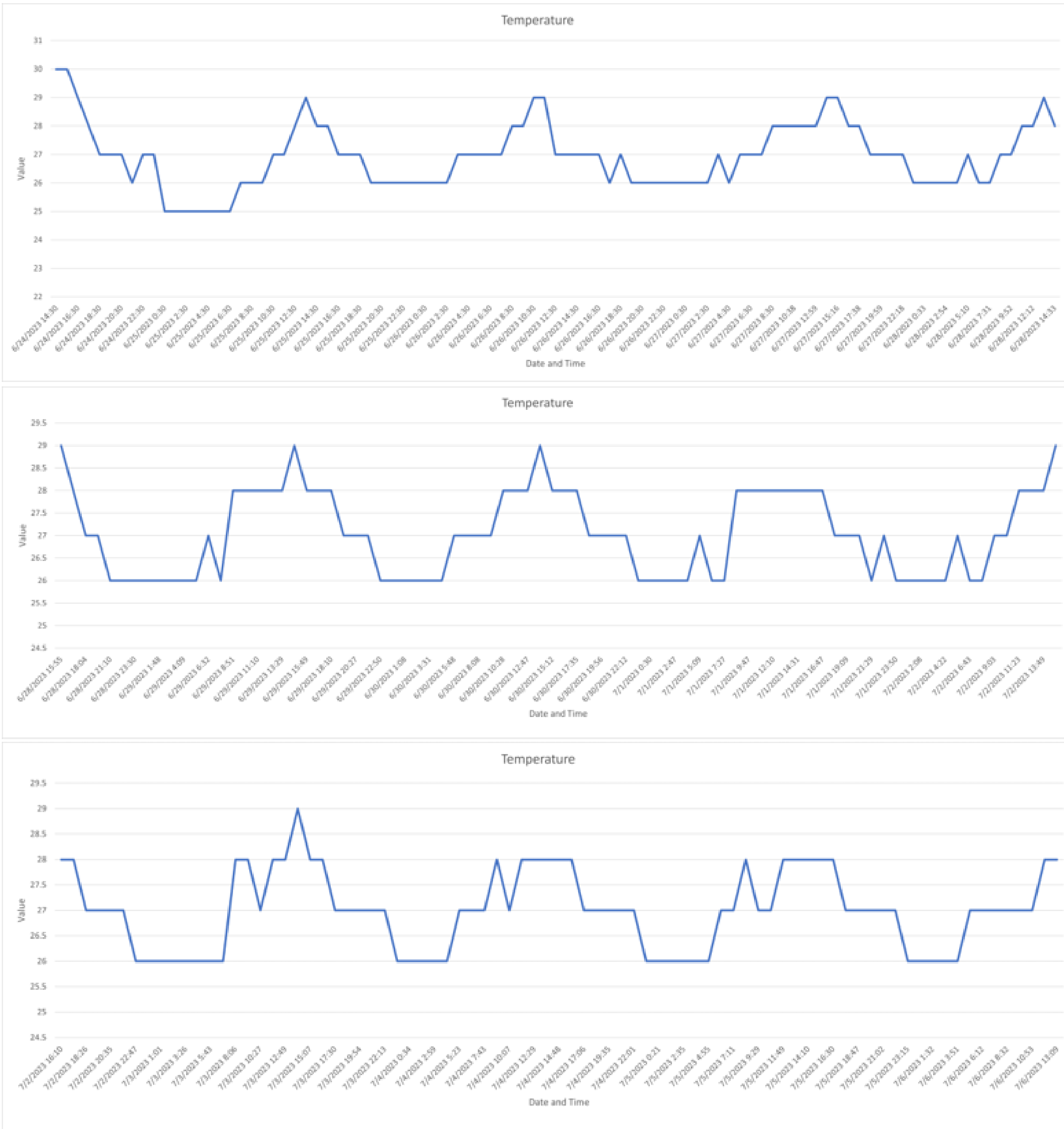


Figure 4.19 *Graphical Representation of Gathered Data for Temperature: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

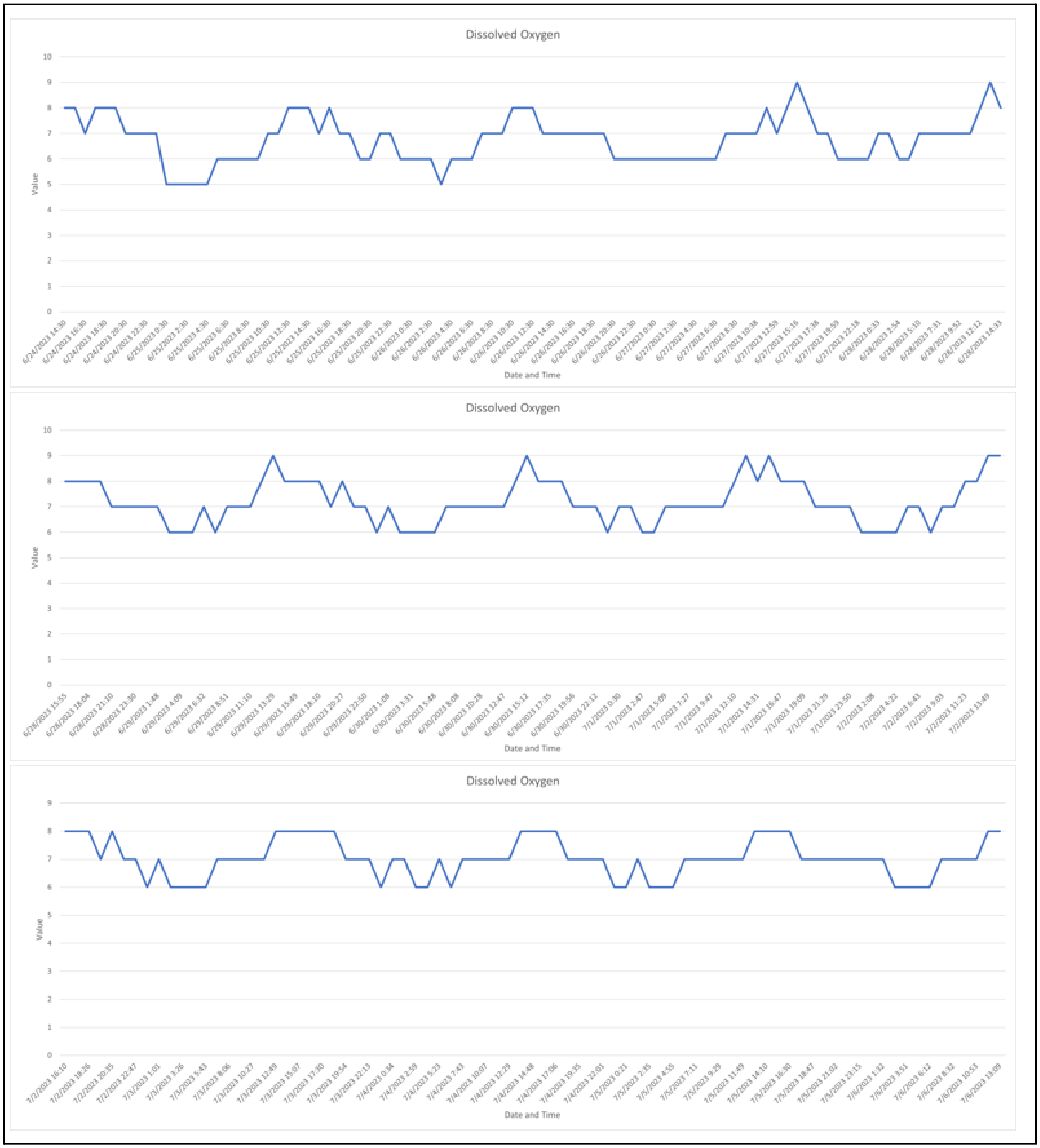


Figure 4.20 *Graphical Representation of Gathered Data for Dissolved Oxygen: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

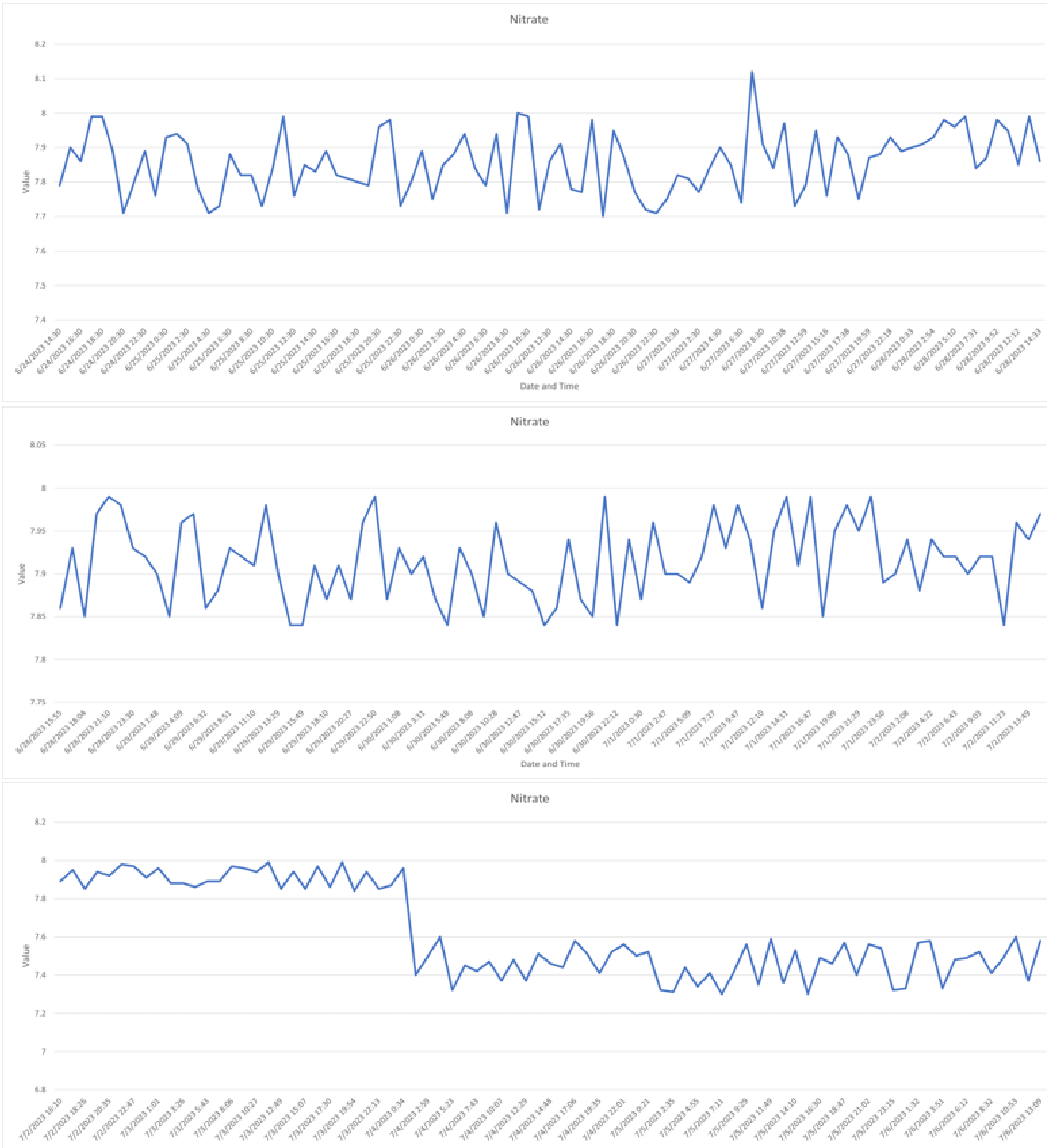


Figure 4.21 *Graphical Representation of Gathered Data for Nitrate: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

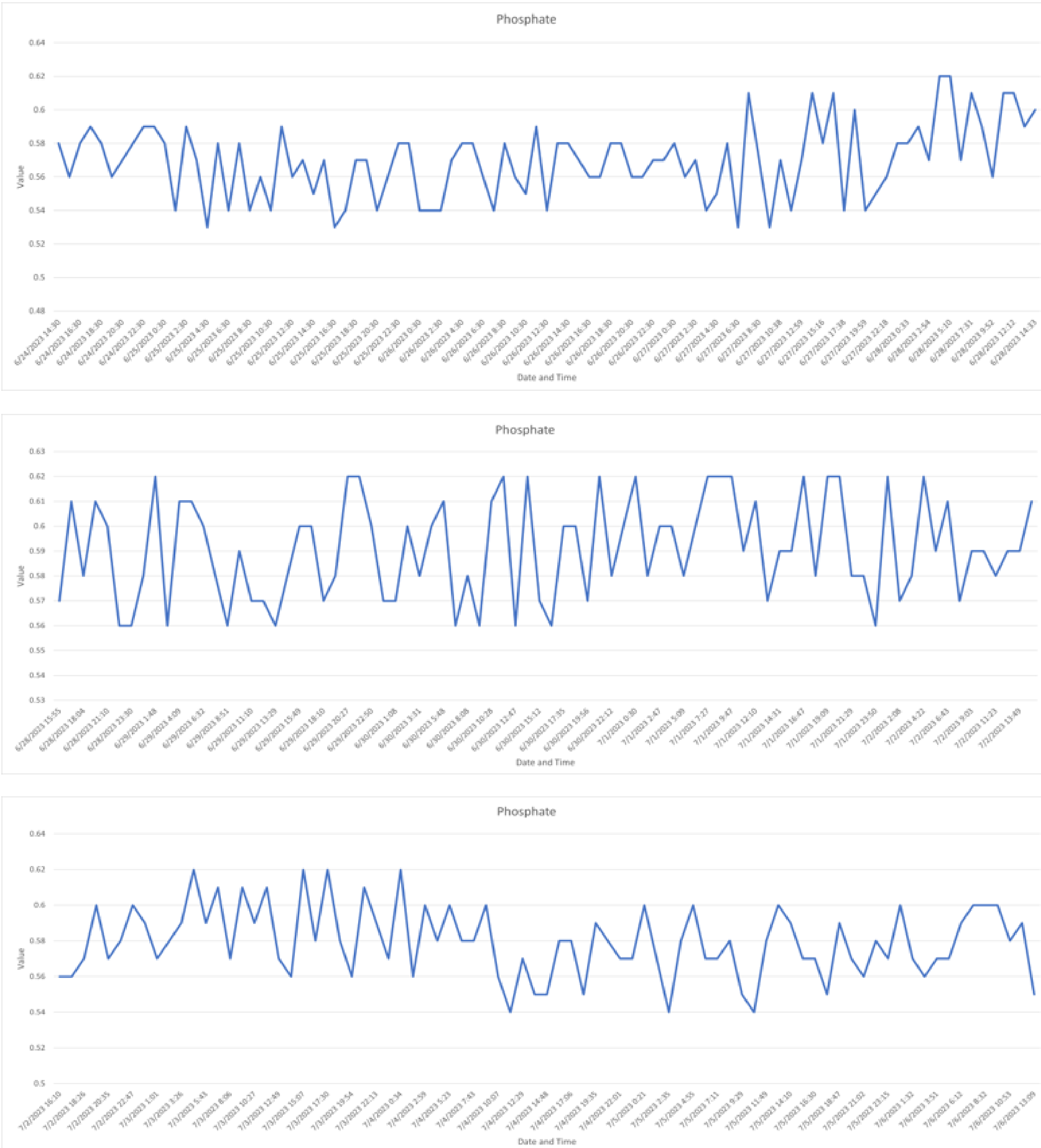


Figure 4.22 *Graphical Representation of Gathered Data for Phosphate: (Top) 1st to 5th day, (Middle) 6th to 9th day, and (Bottom) 10th to 13th day*

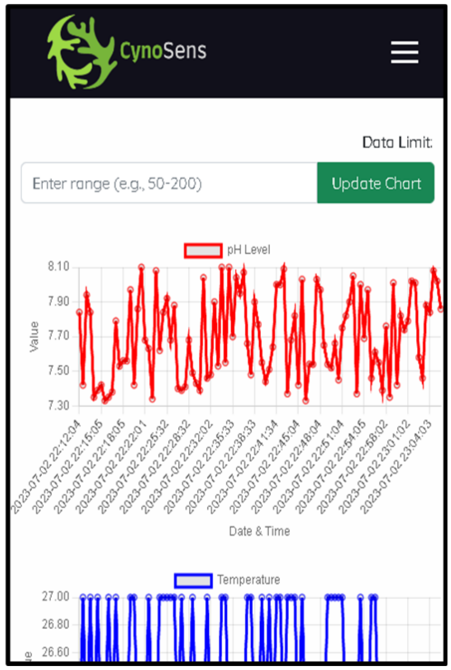


Figure 4.23 *Chart (Smartphone View)*

Figures 4.17 to 4.23 depicted graphical representations of real-time collected data that were displayed on the webpage. The webpage presented a chart that portrayed the values of six (6) water quality parameters, each with its respective chart. The data was exhibited on the y-axis, while the x-axis denoted the corresponding date and time. Additionally, an algae content chart was integrated to visually depict trends and fluctuations in the algae level within the water. Notably, an interactive feature was implemented allowing users to assess the algae level by modifying the color of pointers, based on the legend displayed below the algae content chart. The legend provided color classifications such as blue for low algae levels, green for normal levels, yellow for above-normal levels, and red to indicate the presence of harmful levels of algae. It is worth mentioning that the data was presented in real-time, ensuring accuracy by automatically updating every minute. To facilitate effortless navigation, users could effortlessly hover their mouse cursor on a computer or use their fingers on smartphones to explore specific data points and view the corresponding values. Moreover, users were granted the flexibility to personalize their chart display by selecting a desired data range, choosing which data to be displayed, including or excluding specific information, and even zooming in or out on their smartphones. The amalgamation of real-time updates, interactive features, and user-friendly customization options rendered this chart webpage an exceptional tool for monitoring and analyzing water quality data.

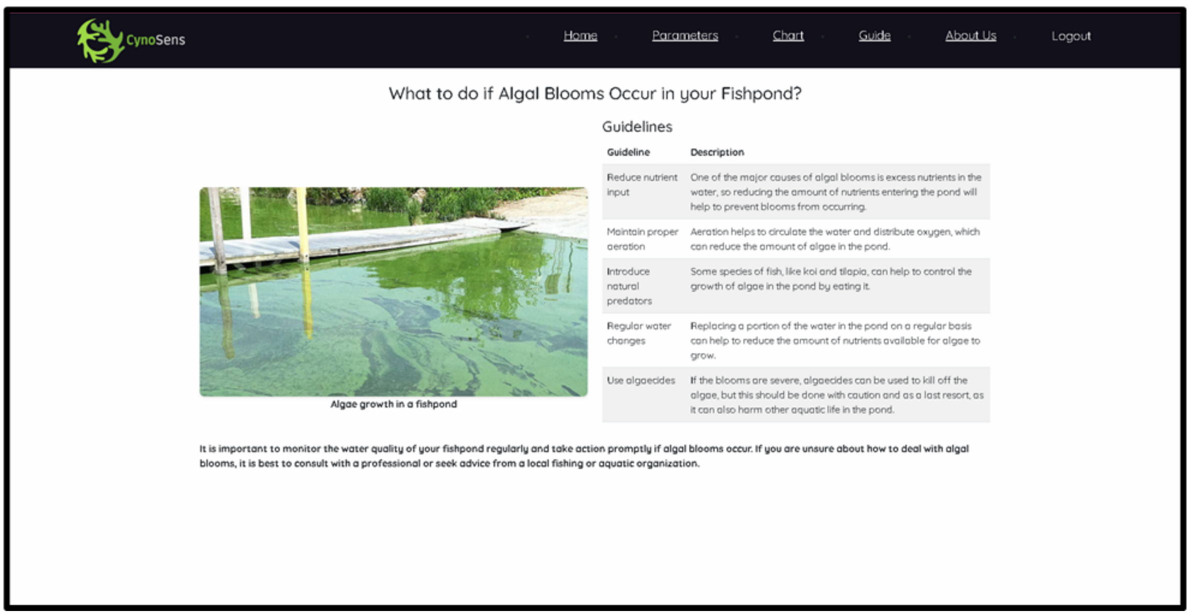


Figure 4.24 *Guide (Computer View)*

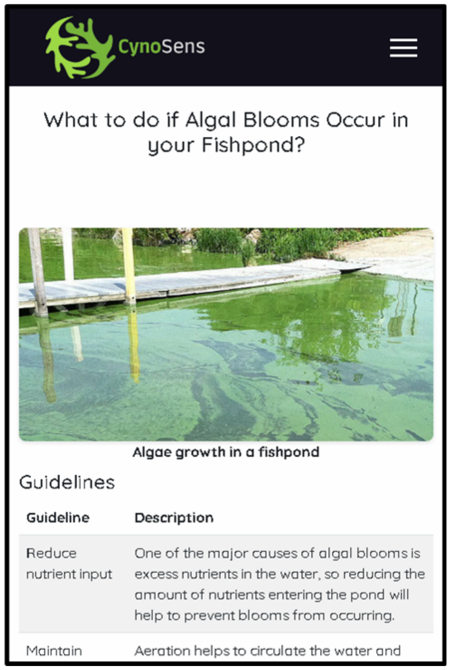


Figure 4.25 *Guide (Smartphone View)*

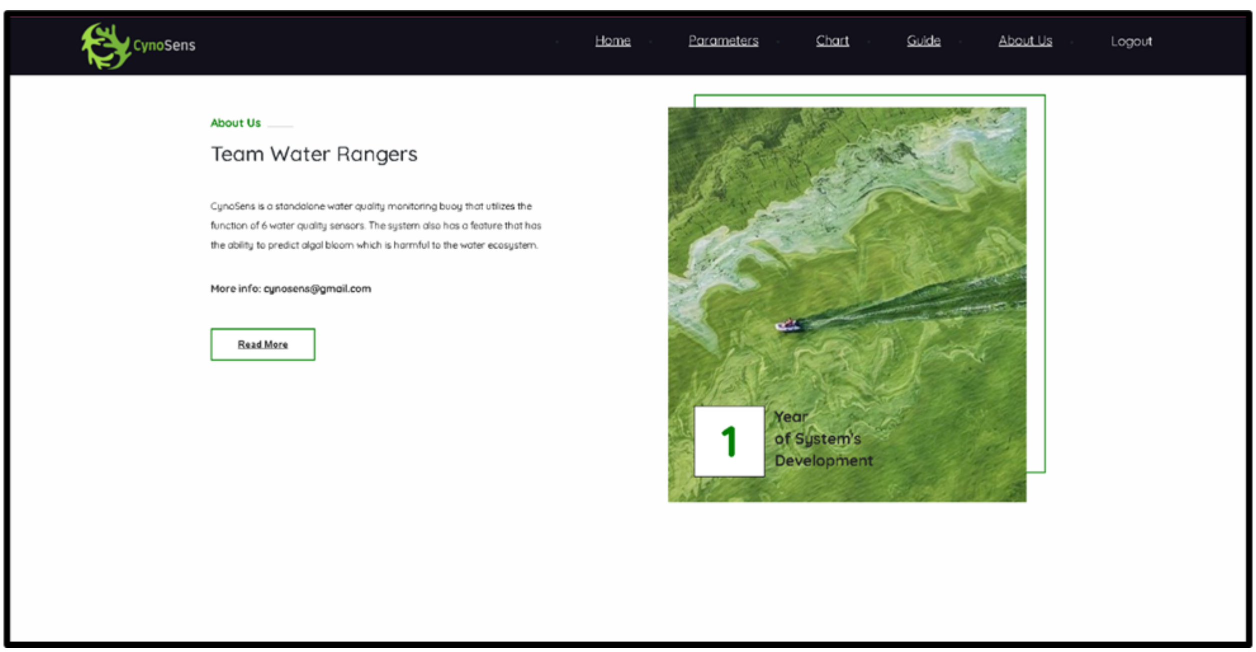


Figure 4.26 *About Us (Computer View)*

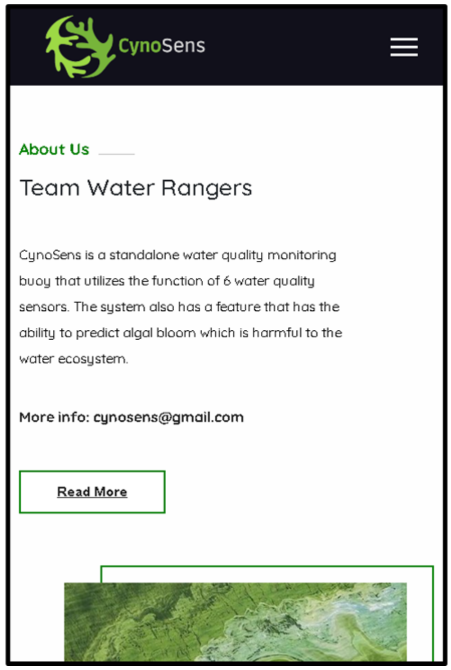


Figure 4.27 *About Us (Smartphone View)*

Chapter 5

SUMMARY OF FINDINGS, CONCLUSION, AND RECOMMENDATIONS

This chapter contains the summary of the study, its findings, conclusions drawn, and presents recommendations for utilization and further research and investigation.

5.1. Summary

Harmful algal blooms (HABs) seriously threaten aquatic ecosystems and human health. Effective management and mitigation techniques depend on early HAB prediction and monitoring. A network of interconnected sensors placed in bodies of water such as lakes and fishponds make up the IoT-based Harmful Algal Bloom Prediction System. These sensors gathered information about the lake's temperature, pH, dissolved oxygen, conductivity, nitrate, and phosphate. The sensors utilized microcontrollers with wireless communication protocols to transmit the data they gathered to a web database.

The proposed device utilized Random Forest Regression as the machine learning algorithm that combines multiple decision trees to make predictions. It is capable of handling complex and non-linear relationships between input variables and the target variable. In the context of harmful algal bloom prediction, RFR utilizes historical water quality data and associated algal bloom occurrences to train a model that can forecast future bloom events. The gathered data is then transmitted to the database, which is then displayed to the web application where users can view the readings.

The proposed device shows better functionality compared to the other water quality monitoring systems available in the market. Other devices such as the Sonde YSI Monitoring Device monitor water quality but lack wireless capability compared to the proposed project that transmits the data gathered wirelessly to the smartphones or laptops of the fish pond owners. Moreover, nutrient contents of the water such as nitrate and phosphorus are usually tested once a month. With the proposed device, nutrient contents of the water are monitored from time to time with the inclusion of algal bloom prediction through water parameters which is not yet available from currently available products in the market.

5.2. Conclusion

The researchers have concluded that the study, "IoT Based Harmful Algal Bloom Prediction System using Random Forest Regression through Water Quality Parameters," presented efficient data in monitoring and reading water quality parameters through the use of sensors that aids in the prediction of occurring algal blooms. Based on the results and findings gathered on the research, the following conclusions were made:

1. The researchers established a standalone algal bloom monitoring buoy that utilizes water quality sensors namely: Dissolved Oxygen (DO) sensor, temperature sensor, pH level sensor, conductivity sensor, nitrate sensor, and phosphate sensor. The data collected by the standalone algal bloom monitoring buoy is comparable to laboratory testing, which is the existing water quality monitoring method.

2. The development of a program that predicts the harmful algal bloom is established by using the machine learning algorithm Random Forest Regression. Its real-time data collection enables early detection and response to potential bloom events. The modeling approach of Random Forest Regression ensures reliable and adaptable predictions for various water bodies and environmental conditions. The calculated accuracy of the model resulted in a score of 0.9148247255739806, thus being 91.48% accurate.

3. The website application was also successfully developed using HTML for the front end, CSS and Javascript for the website layout and design. As for the backend, the proponents utilized the PHP server programming language to connect the website to the MySQL database. These programming languages are utilized to display the acquired data from the buoy. The website serves as a centralized platform that offers valuable information, data visualization, and user-friendly tools to enhance the understanding and management of algal blooms in water bodies. And with its notification, users will always be updated on any changes to the status of the environment.

4. The functionality of the device was proven to operate effectively and demonstrated significant durability through simulations and actual field testing in a fishpond in BFAR Batangas and Laguna Lake. The system was able to accurately measure the level of water quality parameters and detect the level of algae content present in the water greater than 50 μg/L. The algorithm's prediction was triggered at 21 μg/L with an accuracy of 91.48%.

5.3. Recommendations

Based on the findings and conclusions of this project, the following recommendations are provided:

1. Develop a cost-efficient chlorophyll-a sensor as it is one of the important sensors that can accurately detect harmful algal bloom.

2. Utilizing a mobile app instead of a webpage. Mobile apps offer offline access and the ability to store data locally on the device, enabling users to access and interact with the app even without an internet connection.

3. To further augment the device by developing a system that can automatically correct the water parameters and is not dependent on human intervention, enabling it to predict algal blooms and actively mitigate their occurrence.

References:

2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) in SearchWorks catalog. (2019). Stanford.edu. https://searchworks.stanford.edu/view/13728029

Administrator. (2016, November 20). *Arduino Introduction*. ElectronicsHub. https://www.electronicshub.org/arduino-introduction/

Agarwal, T. (2019, August). *Arduino Mega 2560 Board: Specifications, and Pin Configuration*. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/arduino-mega-2560-board/

Ahmed, U., Mumtaz, R., Anwar, H., Shah, A. A., Irfan, R., & José García-Nieto. (2019). *Efficient Water Quality Prediction Using Supervised Machine Learning*. *11*(11), 2210–2210. https://doi.org/10.3390/w11112210

Akhter, F., Hasin Reza Siddiquei, Alahi, E. E., & Subhas Chandra Mukhopadhyay. (2021). *Recent Advancement of the Sensors for Monitoring the Water Quality Parameters in Smart Fisheries Farming*. *10*(3), 26–26. https://doi.org/10.3390/computers10030026

Altynay Kaidarova, Marengo, M., Marinaro, G., Geraldi, N. R., Duarte, C. M., & Kosel, J. (2018). *Flexible and Biofouling Independent Salinity Sensor*. *5*(23), 1801110–1801110. https://doi.org/10.1002/admi.201801110

Anderson, D. M. (2009). *Approaches to monitoring, control and management of harmful algal blooms (HABs)*. *52*(7), 342–347. https://doi.org/10.1016/j.ocecoaman.2009.04.006

Anderson, J. M., Asche, F., & Garlock, T. (2019). *Economics of Aquaculture Policy and Regulation*. Annual Reviews. https://www.annualreviews.org/doi/10.1146/annurev-resource-100518-093750

Ayele, H., & Minaleshewa Atlabachew. (2021). *Review of characterization, factors, impacts, and solutions of Lake eutrophication: lesson for lake Tana, Ethiopia*. *28*(12), 14233–14252. https://doi.org/10.1007/s11356-020-12081-4

B, R. (2018, December 14). *What is MySQL: MySQL Explained For Beginners*. Hostinger Tutorials. https://www.hostinger.com/tutorials/what-is-mysql

Beyeler, R. (2013). *Pocket Wi-Fi*. Moneyland.ch; moneyland.ch. https://www.moneyland.ch/en/pocket-wifi-definition#:~:text=A%20pocket%20Wi%2DFi%20is,can%20connect%20to%20the%20Internet

Borlaza, G. C., Hernandez, C. G., & Cullinane, M. (2023, July 7) *Aquaculture Pollution | Other Environmental Study*. (2018, February 16). Other Environmental Study. https://elibrary.worldbank.org/doi/abs/10.1596/29249

Demetillo, A. T., Japitana, M. V., & Taboada, E. B. (2019). *A system for monitoring water quality in a large aquatic area using wireless sensor network technology*. *29*(1). https://doi.org/10.1186/s42834-019-0009-4

Espaldon, M. V. & et. al. (2005). Ecosystems and People: The Philippine Millennium Ecosystem Assessment (MA) Sub-Global Assessment. *Sub-Global*, 55–196.

Fabiane Fantinelli Franco, Libu Manjakkal, Dhayalan Shakthivel, & Ravinder Dahiya. (2019). *ZnO based Screen Printed Aqueous Ammonia Sensor for Water Quality Monitoring*. https://doi.org/10.1109/sensors43011.2019.8956763

Gravity__Analog_Electrical_Conductivity_Sensor___Meter_V2__K=1__SKU_DFR0300-DFRobot. (2018). Dfrobot.com. https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor___Meter_V2__K=1__SKU_DFR0300

Gravity__Analog_Electrical_Conductivity_Sensor___Meter-DFRobot. (2023). Dfrobot.com. https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor___Meter

Gu, J., Börklü, H. R., Helvacilar, E., & Özdemir, V. (2017). *Conceptual Design of a New Buoy* (Vol. 4, Issue 4)

Harrison, J., Lucius, M. A., Farrell, J. L., Eichler, L. W., & Relyea, R. A. (2021). *Prediction of stream nitrogen and phosphorus concentrations from high-frequency sensors using Random Forests Regression*. *763*, 143005–143005. https://doi.org/10.1016/j.scitotenv.2020.143005

Harrison, J., Lucius, M. A., Farrell, J. L., Eichler, L. W., & Relyea, R. A. (2021). *Prediction of stream nitrogen and phosphorus concentrations from high-frequency sensors using Random Forests Regression*. *763*, 143005–143005. https://doi.org/10.1016/j.scitotenv.2020.143005

Heisler, J., Glibert, P. M., Burkholder, J. M., Anderson, D. Z., Cochlan, W. P., Dennison, W. C., Dortch, Q., Gobler, C. J., Heil, C. A., Humphries, E., Lewitus, A. J., Magnien, R. E., Marshall, H. L., Sellner, K. G., Stockwell, D. A., Stoecker, D. K., & Suddleson, M. (2008). *Eutrophication and harmful algal blooms: A scientific consensus*. *8*(1), 3–13. https://doi.org/10.1016/j.hal.2008.08.006

J.E. Brignell, & Dorey, A. P. (1983). *Sensors for microprocessor-based applications*. https://doi.org/10.1088/0022-3735/16/10/003

Kate, S., Santos, B. S., & Magbanua, F. S. (2020). Geometric Morphometric Analysis of Channa striata (Striped Snakehead) Populations from Laguna de Bay, Philippines Reveals Shape Differences in Relation to Water Quality. *Science Diliman: A Journal of Pure and Applied Sciences*, *32*(2). https://www.journals.upd.edu.ph/index.php/sciencediliman/article/view/7384

Ki Wook Kim, Min Whan Jung, Yiu Fai Tsang, & Kwon, H.-H. (2020). *Stochastic modeling of chlorophyll-a for probabilistic assessment and monitoring of algae blooms in the Lower Nakdong River, South Korea*. *400*, 123066–123066. https://doi.org/10.1016/j.jhazmat.2020.123066

Lauguico, S., Concepcion, R., Rogelio Ruzcko Tobias, Alejandrino, J., Macasaet, D., & Dadios, E. P. (2020). *Indirect Measurement of Dissolved Oxygen Based on Algae Growth Factors Using Machine Learning Models*. https://doi.org/10.1109/r10-htc49770.2020.9357014

Lee, S.-M., & Dong Hyun Lee. (2018). *Improved Prediction of Harmful Algal Blooms in Four Major South Korea's Rivers Using Deep Learning Models*. *15*(7), 1322–1322. https://doi.org/10.3390/ijerph15071322

Lorenzo, A. R., Dula, A. Y., Aldrin, N., John Randolph Munda, Noli, B., Padilla, M., Madrigal, A. M., Amado, T. M., & Karlo, L. (2019). *Dissolved Oxygen (DO) Meter Hydrological Modelling Using Predictive Algorithms*. https://doi.org/10.1109/hnicem48295.2019.9072867

Luo Hongpin, Guanglin, L., Pan Weifeng, Jie, S., & Bai Qiuwei. (2015). *Real-time remote monitoring system for aquaculture water quality*. *8*(6), 136–143. https://doi.org/10.25165/ijabe.v8i6.1486

Lutkevich, B. (2020). *HTML (Hypertext Markup Language)*. TheServerSide.com; TheServerSide.com. https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language

Mapa, D. (n.d.). *REPUBLIC OF THE PHILIPPINES HIS EXCELLENCY PRESIDENT RODRIGO ROA DUTERTE PHILIPPINE STATISTICS AUTHORITY*. Retrieved July 11, 2023, from https://psa.gov.ph/sites/default/files/Fisheries%20Situation%20Report%20for%20Major%20Species%2C%20January%20to%20December%202021.pdf

Maria Shamina D'Silva, Arga Chandrashekar Anil, Naik, R. K., & D'Costa, P. M. (2012). *Algal blooms: a perspective from the coasts of India*. *63*(2), 1225–1253. https://doi.org/10.1007/s11069-012-0190-9

Matías Insausti, Timmis, R., Kinnersley, R., & Rufino, M. C. (2020). *Advances in sensing ammonia from agricultural sources*. *706*, 135124–135124. https://doi.org/10.1016/j.scitotenv.2019.135124

Mell, P. M., & Grance, T. (2011, September 28). *The NIST Definition of Cloud Computing*. NIST. https://www.nist.gov/publications/nist-definition-cloud-computing

Mendoza, M. U., et al. (2019). Water quality and weather trends preceding fish kill occurrences in Lake Taal (Luzon Is., Philippines) and recommendations on its long-term monitoring. *Vol. 12, No. 02*

Menon, G., Maneesha Vinodini Ramesh, & P. Divya. (2017). *A low-cost wireless sensor network for water quality monitoring in natural water bodies*. https://doi.org/10.1109/ghtc.2017.8239341

Nadarajah, S., & Flaaten, O. (2017). *Global aquaculture growth and institutional quality*. *84*, 142–151. https://doi.org/10.1016/j.marpol.2017.07.018

Nag, A., Subhas Chandra Mukhopadhyay, & Kosel, J. (2017). *Sensing system for salinity testing using laser-induced graphene sensors*. *264*, 107–116. https://doi.org/10.1016/j.sna.2017.08.008

Ngatia, L. W., Grace, J. M., Moriasi, D. N., & Taylor, R. W. (2019). *Nitrogen and Phosphorus Eutrophication in Marine Ecosystems*. https://doi.org/10.5772/intechopen.81869

Nitrate Water Testing Sensors - AquaRead. (2020). Aquaread.com. https://www.aquaread.com/sensors/nitrate

O'Grady, B. (2020, February 25). *What is CSS? The Ultimate Intro - Code Institute Global*. Code Institute Global. https://codeinstitute.net/global/blog/what-is-css-and-why-should-i-learn-it/

Prananingtyas, D., Prayogo, & Susanto Rahardja. (2019). *Effect of Different Salinity Level within Water Against Growth Rate, Survival Rate (FCR) of Catfish (Clarias sp.)*. https://doi.org/10.1088/1755-1315/236/1/012035

Rafaela Costa Cruz, Costa, P., Vinga, S., Ludwig Krippahl, & Julia, M. (2021). *A Review of Recent Machine Learning Advances for Forecasting Harmful Algal Blooms and Shellfish Contamination*. *9*(3), 283–283. https://doi.org/10.3390/jmse9030283

Rakibul Islam Chowdhury, Wahid, K. A., Nugent, K., & Baulch, H. M. (2020). *Design and Development of Low-Cost, Portable, and Smart Chlorophyll-A Sensor*. *20*(13), 7362–7371. https://doi.org/10.1109/jsen.2020.2978758

RMBEL (2018, February 1). *Lake Eutrophication - RMBEL*. (2018, February). RMBEL. https://www.rmbel.info/primer/lake-eutrophication/

Robocraze. (2022, August 4). *What is NodeMCU ESP8266: Complete Guide*. Robocraze; Robocraze. https://robocraze.com/blogs/post/what-is-nodemcu-esp8266

Supakorn Harnsoongnoen. (2021). *Metamaterial-Inspired Microwave Sensor for Detecting the Concentration of Mixed Phosphate and Nitrate in Water*. *70*, 1–6. https://doi.org/10.1109/tim.2021.3086901

Tayaban, M. M., Pintor, K. L., & Vital, P. G. (2017). *Detection of potential harmful algal bloom-causing microalgae from freshwater prawn farms in Central Luzon, Philippines, for bloom monitoring and prediction*. *20*(3), 1311–1328. https://doi.org/10.1007/s10668-017-9942-8

Techopedia. (2020, August 7). *What is a Mobile Application? - Definition from Techopedia.*

TechTarget Contributor. (2023). *web application (web app)*. Software Quality; TechTarget. https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app

The Effects: Dead Zones and Harmful Algal Blooms | US EPA. (2013, March 12). US EPA. https://www.epa.gov/nutrientpollution/effects-dead-zones-and-harmful-algal-blooms

Tigor Hamonangan Nasution, Dika, S., Sinulingga, E. P., Kasmir Tanjung, & Lukman Adlin Harahap. (2020). *Analysis of the use of SEN0161 pH sensor for water in goldfish ponds*. *851*(1), 012053–012053. https://doi.org/10.1088/1757-899x/851/1/012053

Tiongson, K. N., & Tamayo-Zafaralla, M. (2018). *View of Water Quality and Seasonal Dynamics of Phytoplankton and Zooplankton in the West Bay of Laguna de Bay, Philippines*. (2023). Uplb.edu.ph. https://ovcre.uplb.edu.ph/journals-uplb/index.php/EDJ/article/view/238/222

Toal, R. (2014, June 10). *What is PHP? Uses & Introduction - Code Institute Global*. Code Institute Global. https://codeinstitute.net/global/blog/what-is-php-programming/

Truong, P. (2019, February 24). *Unraveling the Web of IoT: The Internet of Things - Phu Truong - Medium*. Medium; Medium. https://medium.com/@truong21/unraveling-the-web-of-iot-the-internet-of-things-669c230fa4aa

UTMEL. (2022, June 20). *What is Conductivity Sensor?* Utmel.com; Utmel Electronics. https://www.utmel.com/blog/categories/sensors/what-is-conductivity-sensor

What is the Internet of Things (IoT)? (2020). Oracle.com. https://www.oracle.com/ph/internet-of-things/what-is-iot/

Wilhelm, F. M. (2009). *Pollution of Aquatic Ecosystems I*. 110–119. https://doi.org/10.1016/b978-012370626-3.00222-2

Wong Jun Hong, Shamsuddin, N., Pg Emeroylariffion Abas, Rosyzie Anna Apong, Masri, Z., Hazwani Suhaimi, Gödeke, S., & Muhammad. (2021). *Water Quality Monitoring with Arduino Based Sensors*. *8*(1), 6–6. https://doi.org/10.3390/environments8010006

Wurtsbaugh, W. A., Paerl, H. W., & Dodds, W. K. (2019). *Nutrients, eutrophication and harmful algal blooms along the freshwater to marine continuum*. *6*(5). https://doi.org/10.1002/wat2.1373

Zach Paruch. (2021). Semrush Blog. https://www.semrush.com/blog/javascript/

Zhang, Z., Mao, W., Wang, Z., Tan, X., Wu, F., Wang, D., & Fang, X. (2020). *Development of remote monitoring system for aquaculture water quality based on Internet of Things*. https://doi.org/10.1088/1757-899x/768/5/052033

Zola, A. (2021). *Python*. WhatIs.com; TechTarget. https://www.techtarget.com/whatis/definition/Python

APPENDICES

APPENDIX A

Hardware Source Code

@ Arduino Uno

```
#include "DFRobot_EC.h"
#include <EEPROM.h>
#include <OneWire.h>
#include<DallasTemperature.h>
#include <SoftwareSerial.h>
SoftwareSerial espSerial(5, 6);
String str;
#define One_Wire_Bus 2
OneWire oneWire(One_Wire_Bus);
DallasTemperature sensor(&oneWire);
float tempcelsius;
#define EC_PIN A1
float voltage, ecValue;
DFRobot_EC ec;
// DO code
#include <Arduino.h>
#define DO_PIN A0
#define VREF 5000    // VREF (mv)
#define ADC_RES 1024 // ADC Resolution
// Single-point calibration Mode=0
// Two-point calibration Mode=1
#define TWO_POINT_CALIBRATION 0
//#define READ_TEMP (25) // Current water temperature ℃, Or temperature sensor function
// Single point calibration needs to be filled CAL1_V and CAL1_T
#define CAL1_V (2373) // mv
#define CAL1_T (486)   // ℃
// Two-point calibration needs to be filled CAL2_V and CAL2_T
// CAL1 High temperature point, CAL2 Low temperature point
#define CAL2_V (2338) // mv
#define CAL2_T (479)   // ℃
const uint16_t DO_Table[41] = {
    14460, 14220, 13820, 13440, 13090, 12740, 12420, 12110, 11810, 11530,
    11260, 11010, 10770, 10530, 10300, 10080, 9860, 9660, 9460, 9270,
    9080, 8900, 8730, 8570, 8410, 8250, 8110, 7960, 7820, 7690,
    7560, 7430, 7300, 7180, 7070, 6950, 6840, 6730, 6630, 6530, 6410};
uint8_t Temperaturet;
uint16_t ADC_Raw;
uint16_t ADC_Voltage;
uint16_t DO;
int16_t readDO(uint32_t voltage_mv, uint8_t temperature_c)
{
#if TWO_POINT_CALIBRATION == 0
  uint16_t V_saturation = (uint32_t)CAL1_V + (uint32_t)35 * temperature_c - (uint32_t)CAL1_T * 35;
  return (voltage_mv * DO_Table[temperature_c] / V_saturation);
#else
```

```
  uint16_t V_saturation = (int16_t)((int8_t)temperature_c - CAL2_T) * ((uint16_t)CAL1_V - CAL2_V) /
((uint8_t)CAL1_T - CAL2_T) + CAL2_V;
  return (voltage_mv * DO_Table[temperature_c] / V_saturation);
#endif
}
// pH Code
const int analogInPin = A3;
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10], temp;
void setup()
{
  Serial.begin(115200);
  sensor.begin();
  ec.begin();
  espSerial.begin(115200);
}
void loop()
{
  static unsigned long timepoint = millis();
  if (millis() - timepoint > 1000U) // time interval: 1s
  {
    timepoint = millis();
    voltage = analogRead(EC_PIN) / 1024.0 * 5000; // read the voltage
    sensor.requestTemperatures();
    tempcelsius = sensor.getTempCByIndex(0);      // read your temperature sensor to execute temperature
compensation
    ecValue = ec.readEC(voltage, tempcelsius);     // convert voltage to EC with temperature compensation
  }
  // DO measurement
  {
    Temperaturet = (uint8_t)tempcelsius;
    ADC_Raw = analogRead(DO_PIN);
    ADC_Voltage = uint32_t(VREF) * ADC_Raw / ADC_RES;
    DO = readDO(ADC_Voltage, Temperaturet);
  }
  for (int i = 0; i < 10; i++)
  {
    buf[i] = analogRead(analogInPin);
    delay(10);
  }
  for (int i = 0; i < 9; i++)
  {
    for (int j = i + 1; j < 10; j++)
    {
      if (buf[i] > buf[j])
      {
```

```
      temp = buf[i];
      buf[i] = buf[j];
      buf[j] = temp;
    }} }
  avgValue = 0;
  for (int i = 2; i < 8; i++)
    avgValue += buf[i];
  float pHVol = (float)avgValue * 5.0 / 1024 / 6;
  float phValue = -5.70 * pHVol + 21.34;
  Serial.print("Temperature:");
  Serial.print(tempcelsius, 1);
  Serial.print("^C  EC:");
  Serial.print(ecValue, 2);
  Serial.print("ms/cm DO: ");
  Serial.print(DO);
  Serial.print(" PH= ");
  Serial.println(phValue);
  // Send the received sensor data to the NodeMCU
  String sensorData = "Temperature:" + String(tempcelsius, 1) +
                    "^C  EC:" + String(ecValue, 2) +
                    "ms/cm DO: " + String(DO) +
                    " PH= " + String(phValue);
  espSerial.println(sensorData);
  delay(1000);}
```

@ nodeMCU

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>
// Wi-Fi credentials
const char* ssid = "SSID";
const char* password = "password";
// Cloud server details
const char* serverAddress = "https://cynosens.site";
const int serverPort = 80;
WiFiClient client;
SoftwareSerial espSerial(5, 6); // ESP8266 connection pins (RX, TX)

void setup() {
  Serial.begin(115200);
  espSerial.begin(115200);
  espSerial.setTimeout(1000); // Set timeout for reading from espSerial
  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
```

```
  Serial.println("Connected to WiFi!");
  // Print the IP address
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}
void loop() {
  if (espSerial.available()) {
    char receivedData = espSerial.read();
    switch (receivedData) {
      case 'A': {
        int temperature = espSerial.parseInt();
        Serial.print("Temperature: ");
        Serial.println(temperature);
        sendDataToServer("temperature", temperature);
        break;
      }
      case 'B': {
        int phlevel = espSerial.parseInt();
        Serial.print("pH: ");
        Serial.println(phlevel);
        sendDataToServer("phlevel", phlevel);
        break;
      }
      case 'C': {
        int dissolvedoxygen = espSerial.parseInt();
        Serial.print("Dissolved Oxygen: ");
        Serial.println(dissolvedoxygen);
        sendDataToServer("dissolvedoxygen", dissolvedoxygen);
        break;
      }
      case 'D': {
        int nitrate = espSerial.parseInt();
        Serial.print("Nitrate: ");
        Serial.println(nitrate);
        sendDataToServer("nitrate", nitrate);
        break;
      }
      case 'E': {
        int phosphate = espSerial.parseInt();
        Serial.print("Phosphate: ");
        Serial.println(phosphate);
        sendDataToServer("phosphate", phosphate);
        break;
      }
      case 'F': {
        int conductivity = espSerial.parseInt();
        Serial.print("Conductivity: ");
        Serial.println(conductivity);
```

```
      sendDataToServer("conductivity", conductivity);
      break;
    }
   }
  }
}
void sendDataToServer(const char* parameter, int value) {

  // Create HTTP client object
  HTTPClient http;

  // Send POST request to the cloud server
  String url = String(serverAddress) + "/insert_data.php";
  http.begin(client, url);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  // Create the data payload
  String payload = parameter + String("=") + String(value);

  // Print the data being sent
  Serial.print("Data sent: ");
  Serial.println(payload);

  // Send the POST request and get the response
  int httpResponseCode = http.POST(payload);

  // Check the response code
  if (httpResponseCode > 0) {
    if (httpResponseCode == HTTP_CODE_OK) {
      Serial.println("Data inserted successfully");
    } else {
      Serial.print("Error inserting data. HTTP response code: ");
      Serial.println(httpResponseCode);
      String response = http.getString(); // Get the response from the server
      Serial.println("Response: " + response);
    }
  } else {
    Serial.print("Error in HTTP request. HTTP response code: ");
    Serial.println(httpResponseCode); }


  // End the HTTP connection
  http.end();
  delay(5000); // Adjust the delay as per your requirements}
```

APPENDIX B

Software Source Code

@index.php

```
<!doctype html>
<html lang="en">
<head>
  <title>Index</title>
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
   <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
                                                  <link                          rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.2/font/bootstrap-icons.css">
</head>
<body>
  <section class="vh-100">
<div class="container py-5 h-100">
        <div class="row d-flex align-items-center justify-content-center h-100">
        <div class="col-lg-6">
        <div class="text-center">
        <img src="assets/CynoSensAppLogo.png" class="img-fluid" alt="logo" style="max-height: 300px;
width: auto;">
        </div>
        </div>
        <div class="col-lg-5">
      <form action="login.php" method="post">
        <h1 class="text-center fw-bold mb-4">Login</h1>
        <div class="mb-4">
        <label    class="form-label"    for="form1Example13"><i    class="bi    bi-person-circle"></i>
Username</label>
        <input type="email" id="form1Example13" class="form-control form-control-lg" name="username"
autocomplete="off" placeholder="Enter your email" style="border-radius: 25px;">
        </div>
        <div class="mb-4">
        <label    class="form-label"    for="form1Example23"><i    class="bi    bi-chat-left-dots-fill"></i>
Password</label>
        <input     type="password"     id="form1Example23"     class="form-control     form-control-lg"
name="password" autocomplete="off" placeholder="Enter your password" style="border-radius: 25px;">
        </div>
        <div class="text-center">
        <input type="submit" value="Sign in" name="login" class="btn btn-warning btn-lg text-light px-5"
style="border-radius: 30px; font-weight: 600; background-color: green; border-color: lightgreen;">
        </div>
        </form>
```

```
        <p class="text-center mt-3">Don't have an account? <a href="register.php" class="text-warning"
style="font-weight: 600; text-decoration: none; color: green;">Register Here</a></p>
        </div>
        </div>
</div>
 </section>
 <!-- Bootstrap JavaScript Libraries -->
 <script src="js/popper.min.js"></script>
 <script src="js/bootstrap.min.js"></script>
</body></html>
```

@login.php

```
<?php
session_start();
include_once('connection.php');
if (isset($_POST['login'])) {
        $username = $_POST['username'];
$password = md5($_POST['password']);
        $sql = "SELECT * FROM `tbl_user` WHERE `username`='$username' AND
`password`='$password'";
$result = mysqli_query($conn, $sql);
if (empty($_POST['username']) && empty($_POST['password'])) {
        echo "<script>alert('Please Fill Username and Password');</script>";
        exit;
} elseif (empty($_POST['password'])) {
        echo "<script>alert('Please Fill Password');</script>";
        exit;
} elseif (empty($_POST['username'])) {
        echo "<script>alert('Please Fill Username);</script>";
        exit;
} else {
        if (mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_array($result);
          $name = $row['name'];
        $username = $row['username'];
        $password = $row['password'];
        if ($username == $username && $password == $password) {
        $_SESSION['name'] = $name;
        $_SESSION['username'] = $username;
        $_SESSION['password'] = $password;
        header('location:home.php');
        }
        } else {
        echo "<script>alert('Invalid Username or Password');</script>";
        exit;
        }
}}
```

@Register.php

```
<!doctype html>
<html lang="en">
<head>
  <title>Register</title>
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
                                        <link                       rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.2/font/bootstrap-icons.css">
</head>
<body>
   <section class="vh-100" style="background-color: #eee;">
<div class="container h-100">
            <div class="row d-flex justify-content-center align-items-center h-100">
            <div class="col-lg-12 col-xl-11">
            <div class="card text-black" style="border-radius: 25px;">
            <div class="card-body p-md-2">
            <div class="row justify-content-center">
            <p class="text-center h1 fw-bold mb-4 mx-1 mx-md-3 mt-3">Sign up</p>
            <div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">

                        <form class="mx-1 mx-md-4" action="add.php" method="post">
                        <div class="d-flex flex-row align-items-center mb-4">
                        <i class="fas fa-user fa-lg me-3 fa-fw"></i>
                        <div class="form-outline flex-fill mb-0">
                        <label  class="form-label"  for="form3Example1c"><i  class="bi  bi-person-circle"></i>
Name:</label>
                        <input  type="text"  id="form3Example1c"  class="form-control  form-control-lg  py-3"
name="name" autocomplete="off" placeholder="enter your name" style="border-radius:25px ;" />
                        </div>
                        </div>
                        <div class="d-flex flex-row align-items-center mb-4">
                        <i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
                        <div class="form-outline flex-fill mb-0">
                        <label  class="form-label"  for="form3Example3c"><i  class="bi  bi-envelope-at-fill"></i>
Email:</label>
                        <input  type="email"  id="form3Example3c"  class="form-control  form-control-lg  py-3"
name="username" autocomplete="off" placeholder="enter your username" style="border-radius:25px ;" />
                        </div>
```

```html
                    </div>
                    <div class="d-flex flex-row align-items-center mb-4">
                    <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
                    <div class="form-outline flex-fill mb-0">
                    <label class="form-label" for="form3Example4c"><i class="bi bi-chat-left-dots-fill"></i>
Password:</label>
                           <input type="password" id="form3Example4c" class="form-control form-control-lg
py-3" name="password" autocomplete="off" placeholder="enter your password"
style="border-radius:25px ;" />
                    </div>
                    </div>
                    <div class="d-flex justify-content-center mx-4 mb-3 mb-lg-4">
                    <input type="submit" value="Register" name="register" class="btn btn-warning btn-lg
text-light my-2 py-3" style="width:100% ; border-radius: 30px; font-weight:600; background-color:green ;
border-color: lightgreen ;" />
                    </div>
                    </form>
                    <p align="center">I already have an account <a href="index.php" class="text-warning"
style="font-weight:600; text-decoration:none;">Login</a></p>
          </div>
          <div class="col-md-10 col-lg-6 col-xl-7 d-flex justify-content-center align-items-center">
          <img src="assets/CynoSensAppLogo.png" class="img-fluid" alt="logo" style="max-height: 450px;
width: auto;">
          </div>
          </div>
          </div>
          </div>
          </div>
</div>
 </section>
 <!-- Bootstrap JavaScript Libraries -->
 <script src="js/popper.min.js"></script>
 <script src="js/bootstrap.min.js"></script>
</body></html>
```

@add.php

```php
<?php
include_once('connection.php');
if(isset($_POST['register']))
{
 $name=$_POST['name'];
 $username=$_POST['username'];
 $pass=md5($_POST['password']);
 $sql ="INSERT INTO `tbl_user`(`name`, `username`, `password`) VALUES
('$name','$username','$pass')";
 $result=mysqli_query($conn,$sql);
 if($result){
   header('location:index.php');
```

```
echo"<script>alert('New User Register Success');</script>";
   }else{
           die(mysqli_error($conn)) ;
}}
```

@navigation.php

```
<!-- Global Navigation Bar -->
<link rel="stylesheet" href="css/style.css">
<header>
  <div class="logo"><img src="assets/CynoSensNavBarLogo.png" alt="logo"></div>
  <div class="hamburger">
<div class="line"></div>
<div class="line"></div>
<div class="line"></div>
  </div>
  <nav class="nav-bar">
<ul>
           <li>
           <a href="home.php" <?php if($_SERVER['PHP_SELF'] == 'home.php') echo 'class="active"';
?>>Home</a>
           </li>
           <li>
           <a href="parameters.php" <?php if($_SERVER['PHP_SELF'] == 'parameters.php') echo
'class="active"'; ?>>Parameters</a>
           </li>
           <li>
           <a href="chart.php" <?php if($_SERVER['PHP_SELF'] == 'chart.php') echo 'class="active"';
?>>Chart</a>
           </li>
           <li>
           <a href="guide.php" <?php if($_SERVER['PHP_SELF'] == 'guide.php') echo 'class="active"';
?>>Guide</a>
           </li>
     <li>
           <a href="aboutus.php" <?php if($_SERVER['PHP_SELF'] == 'aboutus.php') echo 'class="active"';
?>>About Us</a>
           </li>
           <li>
           <a href="index.php" class="btn">Logout</a>
           </li>
</ul>
  </nav>
</header>
```

@connection.php

```
<?php
$server = 'localhost';
```

```
$username = 'cynosens;
$password = 'sens';
$database = 'data_db';
if (isset($_POST))
        $conn = new mysqli($server, $username, $password, $database);
if ($conn) {
 // echo 'Server Connected Success';
} else {
 die(mysqli_error($conn));
}
```

@home.php

```
<?php
session_start();
include_once('connection.php');
 // if(isset($_SESSION['name']) && isset($_SESSION['username'] )){
// }
$_SESSION['name'];
$_SESSION['username'];
?>
<!doctype html>
<html lang="en">
 <head>
  <title>Home Page</title>
   <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/home.css">
   <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
 </head>
 <body>
<?php include 'navigation.php'; ?>
<body>
 <div class="slider">
  <div class="slide_viewer">
 <div class="slide_group">
        <div class="slide">
        <img src="assets/image1.jpg" alt="Image 1">
        </div>
        <div class="slide">
        <img src="assets/image2.png" alt="Image 2">
        </div>
        <div class="slide">
```

```
            <img src="assets/image3.png" alt="Image 3">
            </div>
            <div class="slide">
            <img src="assets/image4.jpg" alt="Image 4">
            </div>
            </div>
  </div>
</div><!-- End // .slider -->
<div class="slide_buttons">
</div>
 <div class="directional_nav">
  <div class="previous_btn" title="Previous">
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="65px" height="65px" viewBox="-11 -11.5 65 66">
            <g>
            <g>
            <path fill="#474544"
d="M-10.5,22.118C-10.5,4.132,4.133-10.5,22.118-10.5S54.736,4.132,54.736,22.118
                        c0,17.985-14.633,32.618-32.618,32.618S-10.5,40.103-10.5,22.118z
M-8.288,22.118c0,16.766,13.639,30.406,30.406,30.406
c16.765,0,30.405-13.641,30.405-30.406c0-16.766-13.641-30.406-30.405-30.406C5.35-8.288-8.288,5.35
2-8.288,22.118z"/>
            <path fill="#474544"
d="M25.43,33.243L14.628,22.429c-0.433-0.432-0.433-1.132,0-1.564L25.43,10.051c0.432-0.432,1.132-0
.432,1.563,0
c0.431,0.431,0.431,1.132,0,1.564L16.972,21.647l10.021,10.035c0.432,0.433,0.432,1.134,0,1.564
            c-0.215,0.218-0.498,0.323-0.78,0.323C25.929,33.569,25.646,33.464,25.43,33.243z"/>
            </g>
            </g>
</svg>
  </div>
  <div class="next_btn" title="Next">
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="65px" height="65px" viewBox="-11 -11.5 65 66">
            <g>
            <g>
            <path fill="#474544"
d="M22.118,54.736C4.132,54.736-10.5,40.103-10.5,22.118C-10.5,4.132,4.132-10.5,22.118-10.5
c17.985,0,32.618,14.632,32.618,32.618C54.736,40.103,40.103,54.736,22.118,54.736z M22.118-8.288
c-16.765,0-30.406,13.64-30.406,30.406c0,16.766,13.641,30.406,30.406,30.406c16.768,0,30.406-13.641
,30.406-30.406 C52.524,5.352,38.885-8.288,22.118-8.288z"/>
            <path fill="#474544" d="M18.022,33.569c
0.282,0-0.566-0.105-0.781-0.323c-0.432-0.431-0.432-1.132,0-1.564l10.022-10.035
L17.241,11.615c
0.431-0.432-0.431-1.133,0-1.564c0.432-0.432,1.132-0.432,1.564,0l10.803,10.814c0.433,0.432,0.433,1.1
32,0,1.564 L18.805,33.243C18.59,33.464,18.306,33.569,18.022,33.569z"/>
            </g>
            </g>
```

```
</svg>
  </div>
</div><!-- End // .directional_nav -->
  <!-- jquery -->
  <script src="js/jquery.min.js"></script>
  <!-- Slider JavaScript -->
  <script  src="js/home.js"></script>
  <script>
        hamburger = document.querySelector(".hamburger");
        hamburger.onclick = function() {
        navBar = document.querySelector(".nav-bar");
        navBar.classList.toggle("active");
        }
  </script>
  <!-- Bootstrap JavaScript Libraries -->
  <script src="js/popper.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

@Parameters.php

```
<!doctype html>
<html lang="en">
<head>
  <title>Parameters</title>
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/table.css">
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
  <!-- Toastr CSS -->
  <link rel="stylesheet" href="css/toastr.min.css">
  <style>
  .custom-toast {
    background-color: gray;
  }
</style>
</head>
<body>
<?php include 'navigation.php'; ?>
<div class="table-container">
  <table id="data-table">
    <tr>
      <th><strong>PARAMETERS</strong></th>
```

```
      <th>DATA</th>
    </tr>
  </table>
</div>
<!-- jQuery -->
<script src="js/jquery-3.6.0.min.js"></script>
<!-- JavaScript Dependencies -->
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery-3.6.0.min.js"></script>
<script src="js/toastr.min.js"></script>
<script>
$(document).ready(function() {
  var currentNotification = null; // Variable to store the current notification
  var generalNotification = null; // Variable to store the general notification
  var notificationShown = false; // Flag to track if a notification is already displayed

  function updateTable() {
    $.ajax({
      url: "fetch_data.php",
      type: "GET",
      success: function(response) {
        $("#data-table").html(response);
        checkParameterValues(response);
      },
      error: function(xhr, status, error) {
        console.log(xhr.responseText);
      }
    });
  }
  function checkParameterValues(response) {
    var data = $(response).find("td").map(function() {
      return $(this).text();
    }).get();

    var id = parseFloat(data[1]);
    var phlevel = parseFloat(data[3]);
    var temperature = parseFloat(data[5]);
    var dissolvedoxygen = parseFloat(data[7]);
    var nitrate = parseFloat(data[9]);
    var phosphate = parseFloat(data[11]);
    var conductivity = parseFloat(data[13]);
    var algaecontent = parseFloat(data[15]);
    var status = parseFloat(data[17]);
    var time = parseFloat(data[19]);
    var notifications = [];
    if (phlevel > 9) {
      notifications.push("PH level is high!");
    } else if (phlevel < 6.5) {
```

```
  notifications.push("PH level is low!");
} else {
  notifications.push("PH level is normal.");
}
if (temperature > 31) {
  notifications.push("Temperature is high!");
} else if (temperature < 25) {
  notifications.push("Temperature is low!");
} else {
  notifications.push("Temperature is normal.");
}
if (dissolvedoxygen > 9) {
  notifications.push("Dissolved Oxygen is high!");
} else if (dissolvedoxygen < 5) {
  notifications.push("Dissolved Oxygen is low!");
} else {
  notifications.push("Dissolved Oxygen is normal.");
}
if (nitrate > 9) {
  notifications.push("Nitrate level is high!");
} else if (nitrate < 2) {
  notifications.push("Nitrate level is low!");
} else {
  notifications.push("Nitrate level is normal.");
}
if (phosphate > 0.6) {
  notifications.push("Phosphate is high!");
} else if (phosphate < 0.01) {
  notifications.push("Phosphate is low!");
} else {
  notifications.push("Phosphate is normal.");
}
if (conductivity > 1.5) {
  notifications.push("Conductivity is high!");
} else if (conductivity < 0.5) {
  notifications.push("Conductivity is low!");
} else {
  notifications.push("Conductivity is normal.");
}
if (notifications.length > 0) {
  showNotification(notifications.join("<br>"));
} else {
  hideNotification();
}
if (
    algaecontent >= 6 && algaecontent <= 10
) {
  showGeneralNotification("Low algae presence detected.");
```

```
  } else if (
     algaecontent >= 11 && algaecontent <= 20
  ) {
    showGeneralNotification("Algae presence detected.");

  } else if (
     algaecontent >= 21 && algaecontent <= 25
  ) {
       showGeneralNotification("Algae growth increasing detected! If condition persists, algal bloom is expected within a week.")
  }
   } else if (
     algaecontent >= 25 && algaecontent <= 30
  ) {
       showGeneralNotification("Algae growth increasing detected! If condition persists, algal bloom is expected within 2-3 days.")
  } else if (
     algaecontent > 30
  ) {
    showGeneralNotification("Algal bloom presence detected!");

  } else {
     showGeneralNotification("No algae presence detected.");
  }
}
var currentNotification = null;
var notificationShown = false;
var generalNotification = null;

function showNotification(message) {
  if (!currentNotification && !notificationShown) {
    currentNotification = message;
    toastr.options = {
      positionClass: 'toast-bottom-right',
      timeOut: 3000,
      progressBar: true,
      onHidden: function() {
        currentNotification = null;
        setTimeout(function() {
          notificationShown = false;
        }, 10000);
      }
    };
    // Apply custom color for toastr
    toastr.info(message, '', { 'toastClass': 'custom-toast' });
    notificationShown = true;
  } else if (currentNotification && notificationShown) {
    setTimeout(function() {
      showNotification(message);
```

```
    }, 10000);
  } else if (currentNotification && !notificationShown) {
    currentNotification = message;
    // Update the existing toastr notification with the new message
    toastr.info(message, '', { 'toastClass': 'custom-toast' });
    notificationShown = true;
  }
}
function hideNotification() {
  if (currentNotification) {
    toastr.clear(); // Clear the currently displayed notification
    currentNotification = null;
    notificationShown = false;
  }
}
function showGeneralNotification(message) {
  if (!generalNotification) {
    generalNotification = message;
    toastr.options = {
      positionClass: 'toast-bottom-right',
      timeOut: 5000,  // Display the notification for N seconds
      progressBar: true,
      onHidden: function() {
        generalNotification = null;
      }
    };

    switch (message) {
      case "Low algae presence detected.":
        toastr.success(message); // Green notification color
        break;
      case "Algae presence detected.":
      case "Algae growth increasing detected! If condition persists, algal bloom is expected within a week.":
      case "Algae growth increasing detected! If condition persists, algal bloom is expected within 2-3
days.":
        toastr.warning(message); // Orange notification color
        playNotificationSound(); // Play the notification sound
        break;
      case "Algal bloom presence detected!":
        toastr.error(message); // Red notification color
        playNotificationSound(); // Play the notification sound
        break;
      default:
        toastr.info(message); // Blue notification color
        break;
    }
  }
}
function playNotificationSound() {
```

```
  var audio = new Audio('sound/notification_sound.wav');
  audio.play();
}
function fetchAndUpdateTable() {
  updateTable();
  setTimeout(fetchAndUpdateTable, 30000); // Fetch and update the table every 30 seconds
}
fetchAndUpdateTable(); // Start the fetch and update process
});
</script>
  <script>
      hamburger = document.querySelector(".hamburger");
      hamburger.onclick = function() {
        navBar = document.querySelector(".nav-bar");
        navBar.classList.toggle("active");
      }
  </script>
</body>
</html>
```

@insert_data.php

```
<?php
date_default_timezone_set('Asia/Manila');
 // MySQL database details
$servername = "localhost";
$username = "cynosens";
$password = "sens";
$dbname = "data_db";
$tableName = "tbl_data";
 // Create a connection to the MySQL database
$conn = new mysqli($servername, $username, $password, $dbname);
 // Check the connection
if ($conn->connect_error) {
 die("Connection failed: " . $conn->connect_error);
}
 // Fetch the values from the HTTP POST request
$phlevel = isset($_POST["phlevel"]) ? $_POST["phlevel"] : null;
$temperature = isset($_POST["temperature"]) ? $_POST["temperature"] : null;
$dissolvedoxygen = isset($_POST["dissolvedoxygen"]) ? $_POST["dissolvedoxygen"] : null;
$nitrate = isset($_POST["nitrate"]) ? $_POST["nitrate"] : null;
$phosphate = isset($_POST["phosphate"]) ? $_POST["phosphate"] : null;
$conductivity = isset($_POST["conductivity"]) ? $_POST["conductivity"] : null;

// Check if all required data is present
if ($phlevel !== null && $temperature !== null && $dissolvedoxygen !== null && $nitrate !== null &&
 $phosphate !== null && $conductivity !== null) {
 // Convert the values to floats
 $phlevel = floatval($phlevel);
```

```
$temperature = floatval($temperature);
$dissolvedoxygen = floatval($dissolvedoxygen);
$nitrate = floatval($nitrate);
$phosphate = floatval($phosphate);
$conductivity = floatval($conductivity);

// Prepare the SQL statement
$sql = "INSERT INTO " . $tableName . " (phlevel, temperature, dissolvedoxygen, nitrate, phosphate, conductivity, time) VALUES (?, ?, ?, ?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("dddddds", $phlevel, $temperature, $dissolvedoxygen, $nitrate, $phosphate, $conductivity, $currentTimestamp);
 // Execute the prepared statement
  if ($stmt->execute()) {
    $response = array(
      "Data Transfer" => "success!",
      "message" => "Data inserted successfully!",
      "phlevel" => $phlevel,
      "temperature" => $temperature,
      "dissolvedoxygen" => $dissolvedoxygen,
      "nitrate" => $nitrate,
      "phosphate" => $phosphate,
      "conductivity" => $conductivity,
      "time" => $currentTimestamp
    );
    echo json_encode($response);
  } else {
    $response = array("status" => "error", "message" => "Error inserting data: " . $stmt->error);
    echo json_encode($response);
  }

  // Close the prepared statement
  $stmt->close();
} else {
  $response = array("status" => "error", "message" => "Incomplete data received");
  echo json_encode($response);
}

// Close the database connection
$conn->close();
?>
<script type="text/javascript">
  // Send the data to the server every 30 seconds
  setInterval(function() {
    // Get the current timestamp
    var currentTimestamp = new Date().toISOString().slice(0, 19).replace('T', ' ');

    // Create an HTTP POST request to send the data
```

```
        var xhr = new XMLHttpRequest();
        xhr.open("POST", "ss1122.php", true);
        xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

        // Prepare the data to be sent
        var data = "phlevel=" + encodeURIComponent(phlevel) +
                   "&temperature=" + encodeURIComponent(temperature) +
                   "&dissolvedoxygen=" + encodeURIComponent(dissolvedoxygen) +
                   "&nitrate=" + encodeURIComponent(nitrate) +
                   "&phosphate=" + encodeURIComponent(phosphate) +
                   "&conductivity=" + encodeURIComponent(conductivity) +
                   "&algaecontent=" + encodeURIComponent(algaecontent);

        // Send the HTTP POST request
        xhr.send(data);
    }, 30000); // 30 seconds
</script>
```

@fetch_data.php

```
<?php
$conn = mysqli_connect("localhost", "cynosens ", "sens", "data_db");
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, phlevel, temperature, dissolvedoxygen, nitrate, phosphate, conductivity, algaecontent,
 status, time FROM tbl_data ORDER BY id DESC LIMIT 1";
$result = $conn->query($sql);
 if ($result->num_rows > 0) {
  $tableHTML = "<tr>";
  $tableHTML .= "<th><strong>PARAMETERS</strong></th>";
  $tableHTML .= "<th>DATA</th>";
  $tableHTML .= "</tr>";
   while ($row = $result->fetch_assoc()) {
 $id = $row["id"];
 $phlevel = $row["phlevel"];
 $temperature = $row["temperature"];
 $dissolvedoxygen = $row["dissolvedoxygen"];
 $nitrate = $row["nitrate"];
 $phosphate = $row["phosphate"];
 $conductivity = $row["conductivity"];
 $algaecontent = $row["algaecontent"];
 $status = $row["status"];
 $time = $row["time"];

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>ID Number:</strong></td>";
 $tableHTML .= "<td>" . $id . "</td>";
```

```
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>pH Level:</strong></td>";
 $tableHTML .= "<td>" . $phlevel . "</td>";
 $tableHTML .= "</tr>";
 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Temperature (&deg;C):</strong></td>";
 $tableHTML .= "<td>" . $temperature . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Dissolved Oxygen (mg/L):</strong></td>";
 $tableHTML .= "<td>" . $dissolvedoxygen . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Nitrate (ppm):</strong></td>";
 $tableHTML .= "<td>" . $nitrate . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Phosphate (ppm):</strong></td>";
 $tableHTML .= "<td>" . $phosphate . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Conductivity (ms/cm):</strong></td>";
 $tableHTML .= "<td>" . $conductivity . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Algae Content (ug/L):</strong></td>";
 $tableHTML .= "<td>" . $algaecontent . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Status:</strong></td>";
 $tableHTML .= "<td>" . $status . "</td>";
 $tableHTML .= "</tr>";

 $tableHTML .= "<tr>";
 $tableHTML .= "<td><strong>Date and Time:</strong></td>";
 $tableHTML .= "<td>" . $time . "</td>";
 $tableHTML .= "</tr>";
  }
} else {
  $tableHTML = "<tr><td colspan='2'>0 results</td></tr>";
```

```
}
$conn->close();
 echo $tableHTML;
?>
```

@chart.php

```
<!doctype html>
<html lang="en">
<head>
  <title>Line Graph</title>
  <!-- CSS →
<link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/chart.css">
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
</head>
<body>
  <?php include 'navigation.php'; ?>
<div class="chart-controls">
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <label class="form-label" for="limitInput">Data Limit:</label>
      </div>
      <div class="col-md-6">
        <div class="input-group">
          <input type="text" class="form-control" id="limitInput" placeholder="Enter range (e.g., 50-200)">
                  <button id="updateButton" class="btn btn-success" onclick="updateChartData(); return
 false;">Update Chart</button>
        </div>
      </div>
    </div>
  </div>
</div>

  <script src="js/chart.js"></script>
  <script src="js/jquery.min.js"></script>
  <script src="js/line_graph.js"></script>
  <script>
    const hamburger = document.querySelector(".hamburger");
    hamburger.onclick = function () {
      const navBar = document.querySelector(".nav-bar");
```

```
      navBar.classList.toggle("active");
    }
  </script>

  <!-- Bootstrap JavaScript Libraries -->
  <script src="js/popper.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

@data.php

```
<?php
// Establish a database connection
$servername = 'localhost';
$username = 'cynosens';
$password = 'sens';
$database = 'data_db';
$conn = new mysqli($servername, $username, $password, $database);
 // Check connection
if ($conn->connect_error) {
 die('Connection failed: ' . $conn->connect_error);
}
// Check if the "limit" parameter is set and valid
$range = isset($_GET['limit']) ? $_GET['limit'] : '';
 // Determine the SQL query based on the range values
if ($range !== '') {
   // Split the range into min and max values
    $rangeValues = explode('_', $range); // Change delimiter from "-" to "_"
    $minTime = $rangeValues[0];
    $maxTime = $rangeValues[1];
        $sql = "SELECT time, phlevel, temperature, dissolvedoxygen, nitrate, phosphate, conductivity, algaecontent FROM tbl_data WHERE time >= '$minTime' AND time <= '$maxTime' ORDER BY time ASC";
} else {
        $sql = "SELECT time, phlevel, temperature, dissolvedoxygen, nitrate, phosphate, conductivity, algaecontent FROM (SELECT * FROM tbl_data ORDER BY time DESC LIMIT 100) sub ORDER BY time ASC";
}

// Query the database for the required data
$result = $conn->query($sql);
 // Prepare an array to hold the data
$data = [
 'labels' => [],
 'phlevel' => [],
 'temperature' => [],
 'dissolvedoxygen' => [],
```

```
'nitrate' => [],
'phosphate' => [],
'conductivity' => [],
'algaecontent' => [],
];
 // Fetch the data from the result set and populate the array
while ($row = $result->fetch_assoc()) {
 $data['labels'][] = $row['id'];
 $data['phlevel'][] = $row['phlevel'];
 $data['temperature'][] = $row['temperature'];
 $data['dissolvedoxygen'][] = $row['dissolvedoxygen'];
 $data['nitrate'][] = $row['nitrate'];
 $data['phosphate'][] = $row['phosphate'];
 $data['conductivity'][] = $row['conductivity'];
 $data['algaecontent'][] = $row['algaecontent'];
}
 // Close the database connection
$conn->close();
// Set the response header to indicate JSON content
header('Content-Type: application/json');
// Convert the data array to JSON and send it as the response
echo json_encode($data);
?>
```

@guide.php

```
<!doctype html>
<html lang="en">
 <head>
  <title>About Us</title>
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/guide.css">
   <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
</head>
 <body>
  <?php include 'navigation.php'; ?>
  <div class="container">
 <h3 class="text-center mt-4">What to do if Algal Blooms Occur in your Fishpond?</h3>
 <div class="row">

            <div class="col-sm-6">
            <div class="image-container">
```

```html
<img src="assets/algae.jpg" alt="Algae in a fishpond" class="img-fluid">
<p class="text-center img-caption">Algae growth in a fishpond</p>
</div>
</div>
<div class="col-sm-6">
<h4>Guidelines</h4>
<table class="table table-striped">
<thead>
<tr>
    <th>Guideline</th>
<th>Description</th>
</tr>
</thead>

<tbody>
<tr>
<td>Reduce nutrient input</td>
<td>One of the major causes of algal blooms is excess nutrients in the water, so reducing the amount of nutrients entering the pond will help to prevent blooms from occurring.</td>
</tr>
<tr>
<td>Maintain proper aeration</td>
<td>Aeration helps to circulate the water and distribute oxygen, which can reduce the amount of algae in the pond.</td>
</tr>
<tr>
<td>Introduce natural predators</td>
<td>Some species of fish, like koi and tilapia, can help to control the growth of algae in the pond by eating it.</td>
</tr>
<tr>
<td>Regular water changes</td>
<td>Replacing a portion of the water in the pond on a regular basis can help to reduce the amount of nutrients available for algae to grow.</td>
</tr>
<tr>
<td>Use algaecides</td>
<td>If the blooms are severe, algaecides can be used to kill off the algae, but this should be done with caution and as a last resort, as it can also harm other aquatic life in the pond.</td>
</tr>
</tbody>
</table>
</div>
</div>
<p>It is important to monitor the water quality of your fishpond regularly and take action promptly if algal blooms occur. If you are unsure about how to deal with algal blooms, it is best to consult with a professional or seek advice from a local fishing or aquatic organization.</p>
 </div>
```

```
  <script>
            hamburger = document.querySelector(".hamburger");
            hamburger.onclick = function() {
            navBar = document.querySelector(".nav-bar");
            navBar.classList.toggle("active");
            }
  </script>
   <!-- Bootstrap JavaScript Libraries -->
  <script src="js/popper.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

@aboutus.php

```
<!doctype html>
<html lang="en">
 <head>
  <title>About Us</title>
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/about.css">
   <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <!-- Bootstrap CSS v5.2.1 -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/bootstrap-icons.css">
 </head>
<body>
  <?php include 'navigation.php'; ?>
   <section class="about-section">
 <div class="container">
            <div class="row">
            <!-- Content Column -->
            <div class="content-column col-md-6 col-sm-12">
            <div class="inner-column">
            <div class="sec-title">
            <div class="title">About Us</div>
            <h2>Team Water Rangers</h2>
            </div>
            <div class="text">CynoSens is a standalone water quality monitoring buoy that utilizes the
 function of 6 water quality sensors. The system also has a feature that has the ability to predict algal
 bloom which is harmful to the water ecosystem.</div>
            <div class="email">More info: <span class="theme_color">cynosens@gmail.com</span></div>
            <a href="about.html" class="theme-btn btn-style-three">Read More</a>
            </div>
            </div>
```

```
            <!-- Image Column -->
            <div class="image-column col-md-6 col-sm-12">
            <div class="inner-column">
            <div class="image">
            <img src="assets/about.jpg" alt="">
            <div class="overlay-box">
            <div class="year-box"><span class="number">1</span>Year <br> of System's <br>
Development</div>
            </div>
            </div>
            </div>
            </div>
            </div>
</div>
 </section>
 <script>
            hamburger = document.querySelector(".hamburger");
            hamburger.onclick = function() {
            navBar = document.querySelector(".nav-bar");
            navBar.classList.toggle("active");
            }
 </script>
 </script>
 <!-- Bootstrap JavaScript Libraries -->
 <script src="js/popper.min.js"></script>
 <script src="js/bootstrap.min.js"></script>
</body></html>
```

@style.css

```
/* for navigation bar and hamburger menu */
@font-face {
 font-family: 'Quicksand';
 src: url('fonts/Quicksand/Quicksand-Regular.ttf') format('truetype');
}
 body {
 background: #fefefe;
 font-family: 'Quicksand', sans-serif;
}
 header {
 width: 100%;
 height: 90px;
 background: #11101b;
 display: flex;
 align-items: center;
 justify-content: space-between;
 padding: 0 100px;
}
```

```
 .logo {
  font-size: 28px;
  font-weight: bold;
  color: green;
}
 img {
  width: 200px;
  height: auto;
}
.hamburger {
  display: none;
  position: relative;
  z-index: 9999;
}
 .nav-bar ul {
  display: flex;
}
 .nav-bar ul li a {
  display: block;
  color: #fefefe;
  font-size: 20px;
  padding: 10px 25px;
  border-radius: 50px;
  transition: 0.2s;
  margin: 0 20px;
}
.nav-bar ul li a:hover {
  color: #11101b;
  background: green;
}
 .nav-bar ul li a.active {
  color: #11101b;
  background: green;
}
@media only screen and (max-width: 1320px) {
  header {
 padding: 0 50px;
  }
}
@media only screen and (max-width: 1100px) {
  header {
 padding: 0 30px;
  }
}
 @media only screen and (max-width: 900px) {
  .hamburger {
 display: block;
 cursor: pointer;
```

```
  }
  .hamburger .line {
 width: 30px;
 height: 3px;
 background: #fefefe;
 margin: 6px 0;
  }
  .nav-bar {
 height: 0px;
 position: absolute;
 top: 80px;
 left: 0;
 right: 0;
 width: 100vw;
 background: #11101b;
 transition: 0.5s;
 overflow: hidden;
 z-index: 999;
  }
 .nav-bar.active {
 height: 450px;
  }
   .nav-bar ul {
 display: block;
 width: fit-content;
 margin: 80px auto 0 auto;
 text-align: center;
 transition: 0.5s;
 opacity: 0;
  }
  .nav-bar.active ul {
 opacity: 1;
  }
   .nav-bar ul li a {
 margin-bottom: 12px;
  }
}
.img-responsive {
  max-width: 100%;
  height: auto;
}
```

@home.css

```
/* homepage styles */
html, body {
        background: #F7F5E6;
        height: 100%;
```

```
		margin: 0;
		padding: 0;
		width: 100%;
}
.slider {
		margin: 0 auto;
		max-width: 1200px;
		margin-top: 50px;
}
 .slide_viewer {
		height: 600px;
		overflow: hidden;
		position: relative;
}
 .slide_group {
  height: 100%;
  position: relative;
  width: 100%;
}
 .slide {
		display: none;
		height: 100%;
		position: absolute;
		width: 100%;
}
 .slide:first-child {
		display: block;
}
 .slide img {
		height: 100%;
		object-fit: contain;
		object-position: center;
		width: 100%;
}
 .slide_buttons {
		left: 50%;
		position: absolute;
		transform: translateX(-50%);
		width: fit-content;
}
a.slide_btn {
		color: #474544;
		font-size: 50px;
		margin: 0 0.275em;
		-webkit-transition: all 0.4s ease-in-out;
		-moz-transition: all 0.4s ease-in-out;
		-ms-transition: all 0.4s ease-in-out;
		-o-transition: all 0.4s ease-in-out;
```

```
            transition: all 0.4s ease-in-out;
}
 .slide_btn.active, .slide_btn:hover {
            color: green;
            cursor: pointer;
}
.directional_nav {
            height: 100px;
            margin: 0 auto;
            max-width: 1000px;
            position: relative;
            top: -340px;
  }
 .previous_btn {
            bottom: 0;
            left: 100px;
            margin: auto;
            position: absolute;
            top: 0;
}
 .next_btn {
            bottom: 0;
            margin: auto;
            position: absolute;
            right: 100px;
            top: 0;
  }
 .previous_btn, .next_btn {
            cursor: pointer;
            height: 65px;
            opacity: 0.5;
            -webkit-transition: opacity 0.4s ease-in-out;
            -moz-transition: opacity 0.4s ease-in-out;
            -ms-transition: opacity 0.4s ease-in-out;
            -o-transition: opacity 0.4s ease-in-out;
            transition: opacity 0.4s ease-in-out;
            width: 65px;
}
 .previous_btn:hover, .next_btn:hover {
            opacity: 1;
}
@media only screen and (max-width: 767px) {
            .previous_btn {
              left: 50px;
            }
            .next_btn {
              right: 50px;
            }
```

```
}
```

@table.css

```
/* parameters styles */
.custom-toast {
 background-color: gray;
}
table {
  border-collapse: collapse;
  width: 100%;
  margin-bottom: 20px;
  margin-left: auto;
  margin-right: auto;
}
 th,
td {
 padding: 30px;
 text-align: left;
 font-size: 20px;
}
 th {
  background-color: green;
  color: white;
}
 tr:nth-child(even) {
  background-color: #f2f2f2;
}
tr:hover {
  background-color: #ddd;
}
/* Title styles */
th:first-child,
td:first-child {
  font-weight: bold;
}

/* Additional table styles */
.table-container {
  max-width: 800px;
  margin: 40px auto 20px;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 4px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.table-container h2 {
```

```
  margin-top: 0;
  margin-bottom: 20px;
}
/* Space between header and table */
.header {
  margin-bottom: 40px;
}
 /* Responsive table styles */
@media (max-width: 800px) {
  .table-container {
 overflow-x: auto;
  }
  table {
 width: auto;
  }
}
```

@chart.css

```
.chart-container {
  margin: 0 auto;
  max-width: 1500px;
  position: relative;
  margin-bottom: 20px; /* Add spacing between charts */
  height: 300px; /* Set a fixed height for the chart container */
}
#lineGraphCanvas {
  width: 100%;
}
.container {
  margin: 30px auto;
  max-width: 1500px;
}
.chart-controls {
  margin-top: 30px;
   margin-bottom: 30px;
  justify-content: flex-end;
  align-items: flex-end;
}
.chart-controls .row {
  display: flex;
  justify-content: flex-end;
  align-items: flex-end;
}
.chart-controls .container {
  margin: 0;
}
.chart-controls .col-md-6 {
```

```
  text-align: center;
}
.chart-controls .form-label {
  display: flex;
  justify-content: flex-end;
  align-items: flex-end;
  margin-bottom: 0.5rem;
  margin-right: 2px;
}
.chart-controls #limitInput {
  width: 50px;
  height: 40px;
}
.chart-controls #updateButton {
  width: 130px;
  height: 40px;
}
.chart-controls #limitInput,
.chart-controls #updateButton {
  margin: 0;
}
.custom-legend {
  display: flex;
  justify-content: center;
  margin-top: 10px;
}
.legend-item {
  display: flex;
  align-items: center;
  margin-right: 10px;
}
.legend-color {
  display: inline-block;
  width: 12px;
  height: 12px;
  margin-right: 5px;
}
.low-color {
  background-color: blue;
}
.normal-color {
  background-color:light green;
}
.above-normal-color {
  background-color: yellow;
}
.harmful-color {
  background-color: red;}
```

@guide.css

```
/* guide styles */
.text-center {
  text-align: center;
}
 .row {
  display: flex;
  flex-wrap: wrap;
  margin-right: -15px;
  margin-left: -15px;
}
.col-sm-6 {
  flex: 0 0 100%;
  max-width: 100%;
  padding: 15px;
  align-items: center; /* Added to vertically align content */
}
.text-center {
  text-align: center;
}

ol {
  padding-left: 20px;
}
.container p:last-child {
  margin-bottom: 0;
}
 h3,
p {
  font-weight: bold;
}
 .img-fluid {
  width: 100%;
  height: auto;
  border-radius: 8px; /* Add a border radius for a rounded appearance */
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Add a subtle box shadow */
}
 .image-container {
  display: flex;
  flex-direction: column; /* Update to a column layout */
  justify-content: center;
  align-items: center;
  height: 100%;
  margin-top: 40px;
}
```

@about.css

```
/* about styles */
HTML CSSResult Skip Results Iframe
 .about-section .image-column .inner-column .image img{
  position:relative;
  width:100%;
  height: auto; /* Adjust the height based on the aspect ratio */
  display:block;
}
 .about-section {
 position: absolute;
 top: 0;
 left: 0;
 width: 100%;
 height: 100%;
 z-index: -1;
 padding: 100px 0;
}
.about-section .content-column{
 position:relative;
 margin-bottom:40px;
}
 .about-section .content-column .inner-column{
 position:relative;
 padding-top:50px;
 padding-right:100px;
}
 .about-section .content-column .text{
 position:relative;
 color:#000000;
 font-size:15px;
 line-height:2em;
 margin-bottom:40px;
}
 .about-section .content-column .email{
 position:relative;
 color:#252525;
 font-weight:700;
 margin-bottom:50px;
}
 .about-section .image-column{
 position:relative;
 margin-bottom:50px;
}
 .about-section .image-column .inner-column{
 position:relative;
 padding:40px 30px 55px 0px;
 margin-left:50px;
}
```

```
.about-section .image-column .inner-column:after{
position:absolute;
content:'';
right:0px;
top:20px;
left:40px;
bottom:100px;
z-index:-1;
border:2px solid green;
}
.about-section .image-column .inner-column .image{
position:relative;
}
.about-section .image-column .inner-column .image:before{
position:absolute;
content:'';
left:-50px;
bottom:-50px;
width:299px;
height:299px;
background:url(img/pattern-2.png) no-repeat;
}
.about-section .image-column .inner-column .image img{
position:relative;
width:100%;
display:block;
}
.about-section .image-column .inner-column .image .overlay-box{
position:absolute;
left:40px;
bottom:48px;
}
.about-section .image-column .inner-column .image .overlay-box .year-box{
position:relative;
color:#252525;
font-size:24px;
font-weight:700;
line-height:1.4em;
padding-left:125px;
}
.about-section .image-column .inner-column .image .overlay-box .year-box .number{
position:absolute;
left:0px;
top:0px;
width:110px;
height:110px;
color: green;
font-size:68px;
```

```
 font-weight:700;
 line-height:105px;
 text-align:center;
 background-color:#ffffff;
 border:1px solid #000000;
}
.about-section .btn-style-three:before {
 position: absolute;
 content: '';
 left: 10px;
 top: 10px;
 z-index: -1;
 right: -10px;
 bottom: -10px;
 background: url(img/pattern-1.jpg) repeat;
}
 .about-section .btn-style-three:hover {
 color: #ffffff;
 background: green;
}
 .about-section .btn-style-three {
 position: relative;
 line-height: 24px;
 color: #252525;
 font-size: 15px;
 font-weight: 700;
 background: none;
 display: inline-block;
 padding: 11px 40px;
 background-color: #ffffff;
 text-transform: capitalize;
 border: 2px solid green;
 font-family: 'Arimo', sans-serif;
}
.sec-title2{
 color:#fff;
}
.sec-title {
 position: relative;
 padding-bottom: 40px;
}
 .sec-title .title {
 position: relative;
 color: green;
 font-size: 18px;
 font-weight: 700;
 padding-right: 50px;
 margin-bottom: 15px;
```

```
display: inline-block;
text-transform: capitalize;
}
.sec-title .title:before {
position: absolute;
content: '';
right: 0px;
bottom: 7px;
width: 40px;
height: 1px;
background-color: #bbbbbb;
}
```

@home.js

```
$('.slider').each(function() {
        var $this = $(this);
        var $group = $this.find('.slide_group');
        var $slides = $this.find('.slide');
        var bulletArray = [];
        var currentIndex = 0;
        var timeout;

        function move(newIndex) {
          var animateLeft, slideLeft;

          advance();
          if ($group.is(':animated') || currentIndex === newIndex) {
        return;
          }
          bulletArray[currentIndex].removeClass('active');
          bulletArray[newIndex].addClass('active');
          if (newIndex > currentIndex) {
        slideLeft = '100%';
        animateLeft = '-100%';
          } else {
        slideLeft = '-100%';
        animateLeft = '100%';
          }
          $slides.eq(newIndex).css({
        display: 'block',
        left: slideLeft
          });
          $group.animate({
        left: animateLeft
          }, function() {
        $slides.eq(currentIndex).css({
          display: 'none'
        });
```

```
$slides.eq(newIndex).css({
  left: 0
});
$group.css({
  left: 0
});
currentIndex = newIndex;
  });
}
function advance() {
  clearTimeout(timeout);
  timeout = setTimeout(function() {
if (currentIndex < ($slides.length - 1)) {
  move(currentIndex + 1);
} else {
  move(0);
}
  }, 4000);
}
$('.next_btn').on('click', function() {
  if (currentIndex < ($slides.length - 1)) {
move(currentIndex + 1);
  } else {
move(0);
  }
});

$('.previous_btn').on('click', function() {
  if (currentIndex !== 0) {
move(currentIndex - 1);
  } else {
move(3);
  }
});
$.each($slides, function(index) {
  var $button = $('<a class="slide_btn">&bull;</a>');
  if (index === currentIndex) {
$button.addClass('active');
  }
  $button.on('click', function() {
move(index);
  }).appendTo('.slide_buttons');
  bulletArray.push($button);
});
advance();});
```

@line_graph.js

```
$(function () {
```

```
  updateChartData();
});

const chartConfig = {
  type: 'line',
  options: {
    maintainAspectRatio: false, // Allow the chart to adjust its size
    responsive: true,
    title: {
      display: true,
      text: 'Parameter Trends',
    },
    scales: {
      x: {
        type: 'time', // Set x-axis type to time
        time: {
          displayFormats: {
            minute: 'YYYY-MM-DD HH:mm:ss', // Format for displaying time labels
          },
          tooltipFormat: 'YYYY-MM-DD HH:mm:ss', // Format for tooltip
        },
        title: {
          display: true,
          text: 'Time', // Change the x-axis title to 'Time'
        },
      },
      y: {
        display: true,
        title: {
          display: true,
          text: 'Value',
        },
        ticks: {
          callback: function (value) {
            return value.toFixed(2); // Display decimal value with 2 decimal places
          },
        },
      },
    },
  },
};

const chartContainers = [];

function createChart(labels, data, label, borderColor, xAxisTitle, yAxisTitle, isAlgaeContentChart) {
  const chartContainer = document.createElement('div');
  chartContainer.classList.add('chart-container');
  document.body.appendChild(chartContainer);
```

```
  chartContainers.push(chartContainer);

  const chartCanvas = document.createElement('canvas');
  chartCanvas.classList.add('lineGraphCanvas');
  chartContainer.appendChild(chartCanvas);

  const chartContext = chartCanvas.getContext('2d');

  const backgroundColors = [];

  if (isAlgaeContentChart) {
    const customLegend = document.createElement('div');
    customLegend.classList.add('custom-legend');
    chartContainer.appendChild(customLegend);

    const lowLegendItem = document.createElement('div');
    lowLegendItem.classList.add('legend-item');
    lowLegendItem.innerHTML = '<span class="legend-color low-color"></span>Low (&lt;11)';
    customLegend.appendChild(lowLegendItem);

    const normalLegendItem = document.createElement('div');
    normalLegendItem.classList.add('legend-item');
    normalLegendItem.innerHTML = '<span class="legend-color normal-color"></span>Normal (11-21)';
    customLegend.appendChild(normalLegendItem);

    const aboveNormalLegendItem = document.createElement('div');
    aboveNormalLegendItem.classList.add('legend-item');
    aboveNormalLegendItem.innerHTML = '<span class="legend-color above-normal-color"></span>Above Normal (21-30)';
    customLegend.appendChild(aboveNormalLegendItem);

    const harmfulLegendItem = document.createElement('div');
    harmfulLegendItem.classList.add('legend-item');
    harmfulLegendItem.innerHTML = '<span class="legend-color harmful-color"></span>Harmful (&gt;30)';
    customLegend.appendChild(harmfulLegendItem);

    const styleLink = document.createElement('link');
    styleLink.rel = 'stylesheet';
    styleLink.href = 'chart.css';
    document.head.appendChild(styleLink);

    // Assign background colors based on value ranges
    data.forEach(value => {
      if (value < 11) {
        backgroundColors.push('rgba(0, 0, 255, 1)'); // Low: Blue
      } else if (value >= 11 && value <= 21) {
        backgroundColors.push('rgba(0, 255, 0, 1)'); // Normal: Green
      } else if (value > 21 && value <= 30) {
```

```
      backgroundColors.push('rgba(255, 255, 0, 1)'); // Above Normal:
    } else {
      backgroundColors.push('rgba(255, 0, 0, 1)'); // Harmful: Red
    }
  });
}

const chartOptions = {
  ...chartConfig.options,
  legend: {
    display: !isAlgaeContentChart,
  },
  scales: {
    x: {
      display: true,
      title: {
        display: true,
        text: xAxisTitle,
      },
    },
    y: {
      display: true,
      title: {
        display: true,
        text: yAxisTitle,
      },
      ticks: {
        callback: function (value) {
          return value.toFixed(2);
        },
      },
    },
  },
};

new Chart(chartContext, {
  ...chartConfig,
  options: chartOptions,
  data: {
    labels: labels,
    datasets: [
      {
        label: label,
        data: data,
        borderColor: borderColor,
          backgroundColor: isAlgaeContentChart ? backgroundColors : 'rgba(0, 0, 0, 0)', // Set background
colors for algae content chart
        fill: false,
```

```
      },
    ],
  },
 });
}

function clearChartContainers() {
  chartContainers.forEach((container) => {
    container.parentNode.removeChild(container);
  });
  chartContainers.length = 0;
}

function updateChartData() {
  const limit = $('#limitInput').val();

  $.ajax({
    type: 'GET',
    url: 'data.php',
    data: 'limit=' + limit,
    async: false,
    success: function (data) {
      console.log(data);
      clearChartContainers();
      createChart(data.labels, data.phlevel, 'pH Level', 'red', 'Date & Time', 'Value', false);
      createChart(data.labels, data.temperature, 'Temperature', 'blue', 'Date & Time', 'Value', false);
        createChart(data.labels, data.dissolvedoxygen, 'Dissolved Oxygen', 'orange', 'Date & Time', 'Value',
false);
      createChart(data.labels, data.nitrate, 'Nitrate', 'violet', 'Date & Time', 'Value', false);
      createChart(data.labels, data.phosphate, 'Phosphate', 'gray', 'Date & Time', 'Value', false);
      createChart(data.labels, data.conductivity, 'Conductivity', 'yellow', 'Date & Time', 'Value', false);
      createChart(data.labels, data.algaecontent, 'Algae Content', 'green', 'Date & Time', 'Value', true);
    },
    error: function (data) {
      console.log('error', data);
    },
  });
}

$(function () {
  // Update the chart data initially
  updateChartData();

  // Update the chart data every 1 minute
  setInterval(updateChartData, 60000);
});
```

@algorithm.py

```
import mysql.connector
import pandas as pd
import nbformat
from nbconvert.preprocessors import ExecutePreprocessor
# Establish a connection to the MySQL database
cnx = mysql.connector.connect(
    host='localhost',
    user='cynosens,
    password='sens',
    database='data_db'
)
# Create a cursor object to execute SQL queries
cursor = cnx.cursor()
# Retrieve data from the table
query = "SELECT id, phlevel, temperature, dissolvedoxygen, nitrate, phosphate, conductivity FROM
 tbl_data"
cursor.execute(query)
# Fetch all rows from the result
result = cursor.fetchall()
# Select the first 7 columns from the result
selected_result = [row[:7] for row in result]
# Convert the selected result to a pandas DataFrame
df = pd.DataFrame(selected_result, columns=['id', 'phlevel', 'temperature', 'dissolvedoxygen', 'nitrate',
 'phosphate', 'conductivity'])
# Close the cursor and connection
cursor.close()
cnx.close()
# Load the algorithm notebook file
with open("hab.ipynb", "r") as f:
    nb = nbformat.read(f, as_version=4)
# Execute the notebook with the provided DataFrame
ep = ExecutePreprocessor(timeout=600)
ep.preprocess(nb, {'metadata': {'path': './'}})
# Get the updated DataFrame from the executed notebook
df_updated = nb.cells[-1].outputs[0].data['text/plain']
# Convert the updated DataFrame string to a pandas DataFrame
df_updated = pd.read_csv(pd.compat.StringIO(df_updated), sep='\t')
# Add the "algaelevel" column to the original DataFrame
df['algaelevel'] = df_updated['algaelevel']
# Define a function to map algaelevel to status
def get_status(algaelevel):
    if algaelevel <= 10:
        return "Algae Level: Low"
    elif algaelevel <= 20:
        return "Algae Level: Normal"
    elif algaelevel <= 30:
        return "Algae Level: Above Normal"
    else:
```

```
        return "Algae Level: Harmful"
# Add the "status" column to the original DataFrame
df['status'] = df['algaelevel'].map(get_status)
# Establish a new connection to the MySQL database
cnx = mysql.connector.connect(
    host='localhost',
    user='cynosens',
    password='sens',
    database='data_db'
)
# Create a new cursor object to execute SQL queries
cursor = cnx.cursor()
# Update the "algaelevel" and "status" columns in the table
for _, row in df.iterrows():
        update_query = f"UPDATE tbl_data SET algaelevel = {row['algaelevel']}, status = '{row['status']}'
 WHERE id = {row['id']}"
    cursor.execute(update_query)
# Commit the changes to the database
cnx.commit()
# Close the cursor and connection
cursor.close()
cnx.close()
```

@hab.ipynb


```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "af886f0d",
   "metadata": {},
   "outputs": [],
   "source": [
 "\n",
 "import pandas as pd\n",
 "import numpy as np\n",
 "import matplotlib.pyplot as plt\n",
 "from sklearn.model_selection import train_test_split\n",
 "from sklearn.ensemble import RandomForestRegressor\n",
 "from sklearn.preprocessing import OneHotEncoder\n",
 "from sklearn.pipeline import make_pipeline\n",
 "from sklearn import metrics"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 38,
   "id": "8c8843ba",
```

```
  "metadata": {},
  "outputs": [],
  "source": [
   "df=pd.read_csv(\"Final.csv\")"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 39,
  "id": "f8929edf",
  "metadata": {},
  "outputs": [
{
        "data": {
        "text/html": [
        "<div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "          vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "          vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "          text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "          <th></th>\n",
        "      <th>CHLOROPHYLL_ugperl</th>\n",
        "      <th>TempWater_cel</th>\n",
        "          <th>OxygenDiss_mgperl</th>\n",
        "      <th>Phosphorus_mgperl</th>\n",
        "      <th>Conductivity_uspercm</th>\n",
        "          <th>pH</th>\n",
        "      <th>Nitrogen_mgperl</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
    "    <tr>\n",
        "          <th>0</th>\n",
        "          <td>11.80</td>\n",
        "          <td>4.4</td>\n",
        "          <td>0.0</td>\n",
```

```
"        <td>0.0136</td>\n",
"        <td>66.0</td>\n",
"        <td>7.27</td>\n",
"        <td>0.67</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>1</th>\n",
"        <td>1.50</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0360</td>\n",
"        <td>87.0</td>\n",
 "      <td>0.00</td>\n",
"        <td>1.26</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>2</th>\n",
"        <td>0.50</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0246</td>\n",
"        <td>112.0</td>\n",
"        <td>0.00</td>\n",
"        <td>1.03</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>3</th>\n",
"        <td>0.67</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0251</td>\n",
"        <td>123.0</td>\n",
"        <td>0.00</td>\n",
"        <td>1.06</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>4</th>\n",
"        <td>0.55</td>\n",
"      <td>5.8</td>\n",
"        <td>0.0</td>\n",
"        <td>0.0112</td>\n",
"        <td>59.0</td>\n",
"        <td>7.30</td>\n",
"        <td>0.59</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
 "</div>"
```

```
        ],
        "text/plain": [
        "   CHLOROPHYLL_ugperl  TempWater_cel  OxygenDiss_mgperl  Phosphorus_mgperl  \\\n",
        "0                11.80            4.4                0.0             0.0136   \n",
        "1                 1.50          0.0            0.0                0.0360   \n",
        "2                 0.50            0.0                0.0             0.0246   \n",
        "3                 0.67            0.0                0.0             0.0251   \n",
        "4            0.55           5.8                0.0                0.0112   \n",
        "\n",
        "   Conductivity_uspercm         pH  Nitrogen_mgperl  \n",
        "0                 66.0  7.27        0.67  \n",
        "1                 87.0  0.00        1.26  \n",
        "2                 112.0  0.00              1.03  \n",
        "3                 123.0  0.00              1.06  \n",
        "4                 59.0  7.30        0.59  "
        ]
        },
        "execution_count": 39,
        "metadata": {},
        "output_type": "execute_result"
}
  ],
  "source": [
"df.head()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 5,
  "id": "d74f7cff",
  "metadata": {},
  "outputs": [
{
        "data": {
        "text/plain": [
        "648"
        ]
        },
        "execution_count": 5,
        "metadata": {},
        "output_type": "execute_result"
}
  ],
  "source": [
"len(df)"
  ]
 },
 {
```

"cell_type": "code",
"execution_count": 40,
"id": "08d58dbc",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"((647, 6), (647,))"
]
},
"execution_count": 40,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"X = df.drop('CHLOROPHYLL_ugperl',axis=1)\n",
"y=df['CHLOROPHYLL_ugperl']\n",
"\n",
"X.shape, y.shape"
]
},
{
"cell_type": "code",
"execution_count": 41,
"id": "86732695",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"((517, 6), (517,), (130, 6), (130,))"
]
},
"execution_count": 41,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"from sklearn.model_selection import train_test_split\n",
"\n",
"X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)\n",
"\n",
"X_train.shape,y_train.shape, X_test.shape, y_test.shape"
]
},

{
"cell_type": "code",
"execution_count": 42,
"id": "659ae531",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"{'bootstrap': True,\n",
" 'ccp_alpha': 0.0,\n",
" 'criterion': 'squared_error',\n",
" 'max_depth': None,\n",
" 'max_features': 'auto',\n",
" 'max_leaf_nodes': None,\n",
" 'max_samples': None,\n",
" 'min_impurity_decrease': 0.0,\n",
" 'min_samples_leaf': 1,\n",
" 'min_samples_split': 2,\n",
" 'min_weight_fraction_leaf': 0.0,\n",
" 'n_estimators': 100,\n",
" 'n_jobs': None,\n",
" 'oob_score': False,\n",
" 'random_state': None,\n",
" 'verbose': 0,\n",
" 'warm_start': False}"
]
},
"execution_count": 42,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"from sklearn.ensemble import RandomForestRegressor\n",
"\n",
"rf_1=RandomForestRegressor()\n",
"\n",
"rf_1.get_params()"
]
},
{
"cell_type": "code",
"execution_count": 43,
"id": "a519261c",
"metadata": {
"scrolled": false
},

"outputs": [
{
"data": {
"text/plain": [
"RandomForestRegressor()"
]
},
"execution_count": 43,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"rf_1.fit(X_train,y_train)"
]
},
{
"cell_type": "code",
"execution_count": 44,
"id": "d7330d4c",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"RMSE on Training Data5.866243258847429\n"
]
}
],
"source": [
"y_pred_train = rf_1.predict(X_train)\n",
"\n",
"from sklearn.metrics import mean_squared_error\n",
"\n",
"rmse_train = np.sqrt(mean_squared_error(y_train, y_pred_train))\n",
"print('RMSE on Training Data{}'.format(rmse_train))"
]
},
{
"cell_type": "code",
"execution_count": 47,
"id": "3e37f062",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",

```
          "text": [
          "Training Score:0.8847852723489533\n"
          ]
}
  ],
  "source": [
"from sklearn.metrics import r2_score\n",
"\n",
"y_pred_train = rf_1.predict(X_train)\n",
"print('Training Score:{}'.format(r2_score(y_train, y_pred_train)))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 48,
  "id": "418e55a0",
  "metadata": {},
  "outputs": [
{
          "name": "stdout",
          "output_type": "stream",
          "text": [
          "Testing Score:0.2459457290766135\n"
          ]
}
  ],
  "source": [
"y_pred_test = rf_1.predict(X_test)\n",
"print('Testing Score:{}'.format(r2_score(y_test, y_pred_test)))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 49,
  "id": "ef01270a",
  "metadata": {},
  "outputs": [
{
          "data": {
           "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "    .dataframe tbody tr th:only-of-type {\n",
          "        vertical-align: middle;\n",
          "    }\n",
          "\n",
          "    .dataframe tbody tr th {\n",
          "        vertical-align: top;\n",
```

```
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"        <th></th>\n",
"        <th>Prediction</th>\n",
"        <th>Target</th>\n",
"        <th>Residual</th>\n",
"        <th>Differences%</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"        <th>0</th>\n",
"        <td>16.2380</td>\n",
"        <td>4.1</td>\n",
"        <td>-12.1380</td>\n",
"        <td>296.048780</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>1</th>\n",
"        <td>1.9931</td>\n",
"        <td>1.1</td>\n",
"        <td>-0.8931</td>\n",
"        <td>81.190909</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>2</th>\n",
"        <td>5.3900</td>\n",
"        <td>6.7</td>\n",
"        <td>1.3100</td>\n",
"        <td>19.552239</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>3</th>\n",
"     <td>18.3660</td>\n",
"        <td>23.9</td>\n",
"        <td>5.5340</td>\n",
"        <td>23.154812</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>4</th>\n",
"        <td>0.0000</td>\n",
```

```
"       <td>0.0</td>\n",
"       <td>0.0000</td>\n",
"       <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>...</th>\n",
"       <td>...</td>\n",
"       <td>...</td>\n",
"       <td>...</td>\n",
"       <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>125</th>\n",
"       <td>3.7680</td>\n",
"       <td>7.2</td>\n",
"       <td>3.4320</td>\n",
"       <td>47.666667</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>126</th>\n",
"       <td>18.7610</td>\n",
"       <td>10.5</td>\n",
"       <td>-8.2610</td>\n",
"       <td>78.676190</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>127</th>\n",
"       <td>10.1330</td>\n",
"       <td>8.0</td>\n",
"       <td>-2.1330</td>\n",
"       <td>26.662500</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>128</th>\n",
"       <td>11.4193</td>\n",
"       <td>23.7</td>\n",
"       <td>12.2807</td>\n",
"       <td>51.817300</td>\n",
"    </tr>\n",
"    <tr>\n",
"       <th>129</th>\n",
"       <td>3.9690</td>\n",
"       <td>1.5</td>\n",
"       <td>-2.4690</td>\n",
"       <td>164.600000</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
```

```
        "<p>130 rows × 4 columns</p>\n",
        "</div>"
        ],
        "text/plain": [
        "          Prediction  Target  Residual  Differences%\n",
        "0         16.2380          4.1  -12.1380    296.048780\n",
        "1         1.9931  1.1   -0.8931     81.190909\n",
        "2         5.3900  6.7    1.3100     19.552239\n",
        "3         18.3660          23.9    5.5340    23.154812\n",
        "4         0.0000  0.0    0.0000              NaN\n",
        "..        ...        ...       ...            ...\n",
        "125     3.7680     7.2     3.4320  47.666667\n",
        "126     18.7610    10.5   -8.2610          78.676190\n",
        "127     10.1330     8.0   -2.1330          26.662500\n",
        "128     11.4193    23.7   12.2807          51.817300\n",
        "129     3.9690     1.5   -2.4690   164.600000\n",
        "\n",
        "[130 rows x 4 columns]"
        ]
        },
        "execution_count": 49,
        "metadata": {},
        "output_type": "execute_result"
}
  ],
  "source": [
"df_eval = pd.DataFrame(rf_1.predict(X_test), columns=['Prediction'])\n",
"y_test = y_test.reset_index(drop=True)\n",
"df_eval['Target'] = y_test\n",
"\n",
"df_eval['Residual'] = df_eval['Target']-df_eval['Prediction']\n",
"df_eval['Differences%'] = np.absolute(df_eval['Residual']/df_eval['Target']*100)\n",
"df_eval"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 50,
  "id": "b778b04f",
  "metadata": {},
  "outputs": [
{
        "data": {
        "text/html": [
        "<div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "          vertical-align: middle;\n",
```

```
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"        <th></th>\n",
"        <th>Prediction</th>\n",
"        <th>Target</th>\n",
"        <th>Residual</th>\n",
"      <th>Differences%</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"        <th>count</th>\n",
"        <td>130.000000</td>\n",
"        <td>130.000000</td>\n",
"        <td>130.000000</td>\n",
"        <td>125.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>mean</th>\n",
"        <td>7.672127</td>\n",
"        <td>7.934692</td>\n",
"         <td>0.262565</td>\n",
"        <td>inf</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>std</th>\n",
"        <td>8.336887</td>\n",
"        <td>17.384310</td>\n",
"        <td>15.093590</td>\n",
"        <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>min</th>\n",
"        <td>0.000000</td>\n",
"        <td>0.000000</td>\n",
"        <td>-21.638000</td>\n",
"        <td>0.019608</td>\n",
```

```
"    </tr>\n",
"    <tr>\n",
"          <th>25%</th>\n",
"          <td>2.476800</td>\n",
"          <td>1.625000</td>\n",
"          <td>-2.950375</td>\n",
"          <td>19.155385</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"          <td>5.033400</td>\n",
"          <td>3.450000</td>\n",
"          <td>-0.674750</td>\n",
"          <td>52.529412</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>75%</th>\n",
"          <td>9.672000</td>\n",
"          <td>7.575000</td>\n",
"          <td>0.561575</td>\n",
"          <td>157.807692</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>max</th>\n",
"          <td>40.407000</td>\n",
"           <td>177.000000</td>\n",
"          <td>158.905000</td>\n",
"          <td>inf</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"          Prediction         Target    Residual  Differences%\n",
"count  130.000000  130.000000  130.000000     125.000000\n",
"mean   7.672127    7.934692    0.262565         inf\n",
"std       8.336887   17.384310   15.093590         NaN\n",
"min      0.000000         0.000000  -21.638000    0.019608\n",
"25%     2.476800          1.625000   -2.950375    19.155385\n",
"50%     5.033400          3.450000   -0.674750    52.529412\n",
"75%     9.672000          7.575000    0.561575    157.807692\n",
"max     40.407000  177.000000  158.905000                inf"
]
},
"execution_count": 50,
"metadata": {},
"output_type": "execute_result"
```

}
  ],
  "source": [
"df_eval.describe()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 53,
  "id": "7dd8ffe9",
  "metadata": {},
  "outputs": [
{
    "name": "stdout",
    "output_type": "stream",
    "text": [
    "Fitting 3 folds for each of 100 candidates, totalling 300 fits\n"
    ]
},
{
    "data": {
    "text/plain": [
    "RandomizedSearchCV(cv=3, estimator=RandomForestRegressor(), n_iter=100,\n",
    "   n_jobs=-1,\n",
    "   param_distributions={'bootstrap': [True, False],\n",
    "   'max_depth': [10, 20, 30, 40, 50, None],\n",
    "   'max_features': ['auto', 'sqrt'],\n",
    "   'min_samples_leaf': [1, 2, 4],\n",
    "   'min_samples_split': [2, 5, 7],\n",
    "   'n_estimators': [100, 200]},\n",
    "   verbose=8)"
    ]
    },
    "execution_count": 53,
    "metadata": {},
    "output_type": "execute_result"
}
  ],
  "source": [
"from sklearn.model_selection import RandomizedSearchCV\n",
"rf_random = RandomForestRegressor()\n",
"random_grid= {'bootstrap':[True, False],\n",
"   'max_depth':[10, 20, 30, 40, 50, None],\n",
"   'max_features':['auto', 'sqrt'],\n",
"   'min_samples_leaf':[1,2,4],\n",
"   'min_samples_split':[2, 5, 7], \n",
"   'n_estimators':[100, 200]}\n",
"\n",

```
"rf_random = RandomizedSearchCV(estimator = rf_random,\n",
"                              param_distributions=random_grid,\n",
"                              n_iter=100,\n",
"                              cv=3,\n",
"                              verbose=8,\n",
"                              n_jobs=-1)\n",
"\n",
"rf_random.fit(X_train, y_train)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 54,
  "id": "e1af04d3",
  "metadata": {},
  "outputs": [
{
            "data": {
            "text/plain": [
      "(RandomForestRegressor(max_depth=10, max_features='sqrt', min_samples_leaf=2,\n",
            "                       min_samples_split=5, n_estimators=200),\n",
            " {'n_estimators': 200,\n",
            "  'min_samples_split': 5,\n",
            "  'min_samples_leaf': 2,\n",
            "  'max_features': 'sqrt',\n",
            "  'max_depth': 10,\n",
            "  'bootstrap': True},\n",
            " 0.3105205102120588)"
            ]
            },
            "execution_count": 54,
            "metadata": {},
            "output_type": "execute_result"
}
  ],
  "source": [
"rf_random.best_estimator_, rf_random.best_params_, rf_random.best_score_"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 55,
  "id": "c9fb22ed",
  "metadata": {},
  "outputs": [
{
            "data": {
            "text/plain": [
```

"9.38767836345566"
]
},
"execution_count": 55,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"np.sqrt(mean_squared_error(y_train, rf_random.predict(X_train)))"
]
},
{
"cell_type": "code",
"execution_count": 57,
"id": "9961d3ba",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"15.002649268192275"
]
},
"execution_count": 57,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"np.sqrt(mean_squared_error(y_test, rf_random.predict(X_test)))"
]
},
{
"cell_type": "code",
"execution_count": 59,
"id": "2fb0311c",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Training r2_score: 0.7049441264298039\n",
"Testing r2_score: 0.2494586357486126\n"
]
}
],

```
  "source": [
"from sklearn.metrics import r2_score\n",
"\n",
"print(f'Training r2_score: {r2_score(y_train,rf_random.predict(X_train))}')\n",
"print(f'Testing r2_score: {r2_score(y_test,rf_random.predict(X_test))}')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 60,
  "id": "9cf8fcfa",
  "metadata": {},
  "outputs": [
{
	"data": {
	"text/html": [
	 "<div>\n",
	"<style scoped>\n",
	"    .dataframe tbody tr th:only-of-type {\n",
	"        vertical-align: middle;\n",
	"    }\n",
	"\n",
	"    .dataframe tbody tr th {\n",
	"        vertical-align: top;\n",
	"    }\n",
	"\n",
	"    .dataframe thead th {\n",
	"        text-align: right;\n",
	"    }\n",
	"</style>\n",
	"<table border=\"1\" class=\"dataframe\">\n",
	"  <thead>\n",
	"    <tr style=\"text-align: right;\">\n",
	"        <th></th>\n",
	"        <th>Prediction</th>\n",
	"        <th>Target</th>\n",
	"        <th>Residual</th>\n",
	"      <th>Differences%</th>\n",
	"    </tr>\n",
	"  </thead>\n",
	"  <tbody>\n",
	"    <tr>\n",
	"        <th>0</th>\n",
	"        <td>8.947569</td>\n",
	"        <td>4.1</td>\n",
	"        <td>-4.847569</td>\n",
	"        <td>118.233393</td>\n",
	"    </tr>\n",
```

```
"    <tr>\n",
"          <th>1</th>\n",
"          <td>1.552601</td>\n",
"          <td>1.1</td>\n",
"          <td>-0.452601</td>\n",
"          <td>41.145573</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>2</th>\n",
 "      <td>5.163621</td>\n",
"          <td>6.7</td>\n",
"          <td>1.536379</td>\n",
"          <td>22.931026</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>3</th>\n",
"          <td>17.633567</td>\n",
"           <td>23.9</td>\n",
"          <td>6.266433</td>\n",
"          <td>26.219385</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>4</th>\n",
"          <td>0.704714</td>\n",
"          <td>0.0</td>\n",
"          <td>-0.704714</td>\n",
"          <td>inf</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>...</th>\n",
"          <td>...</td>\n",
"          <td>...</td>\n",
"          <td>...</td>\n",
"          <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"          <th>125</th>\n",
"          <td>5.386674</td>\n",
"          <td>7.2</td>\n",
"          <td>1.813326</td>\n",
"          <td>25.185079</td>\n",
"    </tr>\n",
"    <tr>\n",
 "      <th>126</th>\n",
"          <td>16.525193</td>\n",
"          <td>10.5</td>\n",
"          <td>-6.025193</td>\n",
"          <td>57.382793</td>\n",
```

```
        "   </tr>\n",
        "   <tr>\n",
        "         <th>127</th>\n",
        "         <td>6.986616</td>\n",
        "         <td>8.0</td>\n",
        "         <td>1.013384</td>\n",
        "         <td>12.667297</td>\n",
        "   </tr>\n",
        "   <tr>\n",
        "         <th>128</th>\n",
        "         <td>11.993982</td>\n",
        "         <td>23.7</td>\n",
        "         <td>11.706018</td>\n",
        "         <td>49.392482</td>\n",
        "   </tr>\n",
        "   <tr>\n",
        "         <th>129</th>\n",
        "         <td>3.955537</td>\n",
        "         <td>1.5</td>\n",
        "         <td>-2.455537</td>\n",
        "         <td>163.702460</td>\n",
        "   </tr>\n",
        "  </tbody>\n",
        "</table>\n",
        "<p>130 rows × 4 columns</p>\n",
        "</div>"
        ],
        "text/plain": [
        "         Prediction  Target   Residual  Differences%\n",
        "0        8.947569         4.1  -4.847569   118.233393\n",
        "1        1.552601         1.1  -0.452601   41.145573\n",
        "2        5.163621         6.7   1.536379   22.931026\n",
        "3        17.633567        23.9   6.266433             26.219385\n",
        "4        0.704714         0.0  -0.704714             inf\n",
        "..        ...         ...        ...           ...\n",
        "125     5.386674     7.2   1.813326        25.185079\n",
        "126   16.525193    10.5  -6.025193        57.382793\n",
        "127     6.986616     8.0   1.013384        12.667297\n",
        "128   11.993982    23.7  11.706018       49.392482\n",
        "129     3.955537     1.5  -2.455537        163.702460\n",
        "\n",
        "[130 rows x 4 columns]"
        ]
        },
        "execution_count": 60,
        "metadata": {},
        "output_type": "execute_result"
}
```

],
"source": [
"df_eval = pd.DataFrame(rf_random.predict(X_test), columns=['Prediction'])\n",
"y_test = y_test.reset_index(drop=True)\n",
"df_eval['Target'] = y_test\n",
"\n",
"df_eval['Residual'] = df_eval['Target']-df_eval['Prediction']\n",
"df_eval['Differences%'] = np.absolute(df_eval['Residual']/df_eval['Target']*100)\n",
"df_eval"
]
},
{
"cell_type": "code",
"execution_count": 76,
"id": "e120c4fa",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"R2 score : 0.2494586357486126\n"
]
}
],
"source": [
"from sklearn.metrics import r2_score\n",
"r = r2_score(y_test, predictions)\n",
"print(\"R2 score : {}\" . format(r))"
]
},
{
"cell_type": "code",
"execution_count": 80,
"id": "c41379ed",
"metadata": {},
"outputs": [],
"source": [
"import pickle\n",
"# open a file, where you ant to store the data\n",
"filename = 'RFR'\n",
"\n",
"# dump information to that file\n",
"pickle.dump(rf_1,open(filename, 'wb'))"
]
},
{
"cell_type": "code",

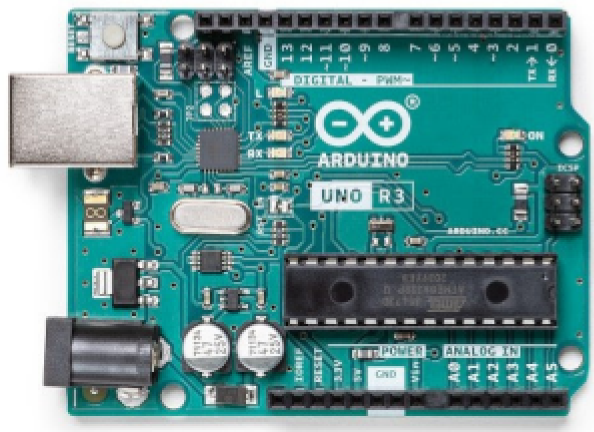
"execution_count": 83,
"id": "071e4b89",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"['RFR1.pkl']"
]
},
"execution_count": 83,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"import joblib\n",
"\n",
"joblib.dump(rf_1, 'RFR1.pkl', compress=3)"
]
},
{
"cell_type": "code",
"execution_count": 81,
"id": "c79125b8",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"'C:\\\\Users\\\\nepteam'"
]
},
"execution_count": 81,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"pwd"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "e34213f6",
"metadata": {},
"outputs": [],

```
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
"name": "ipython",
"version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.13"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

APPENDIX C

Device Specification Sheet

Arduino Uno V3



Arduino Uno V3 Specifications & Features

- **ATMega328P** Processor
  - **Memory**
    - **AVR CPU at up 16 MHz**
    - **32KB Flash**
    - **2KB SRAM**
    - **1KB EEPROM**
- **Security**
  - Power on Reset (POR)
  - Brown Out Detection (BOD)
- **Peripherals**
  - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
  - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
  - 1x USART with fractional baud rate generator and start-of-frame detection
  - 1x controller/peripheral Serial Peripheral Interface (SPI)
  - 1x Dual mode controller/peripheral I2C
  - 1x Analog Comparator (AC) with a scalable reference input

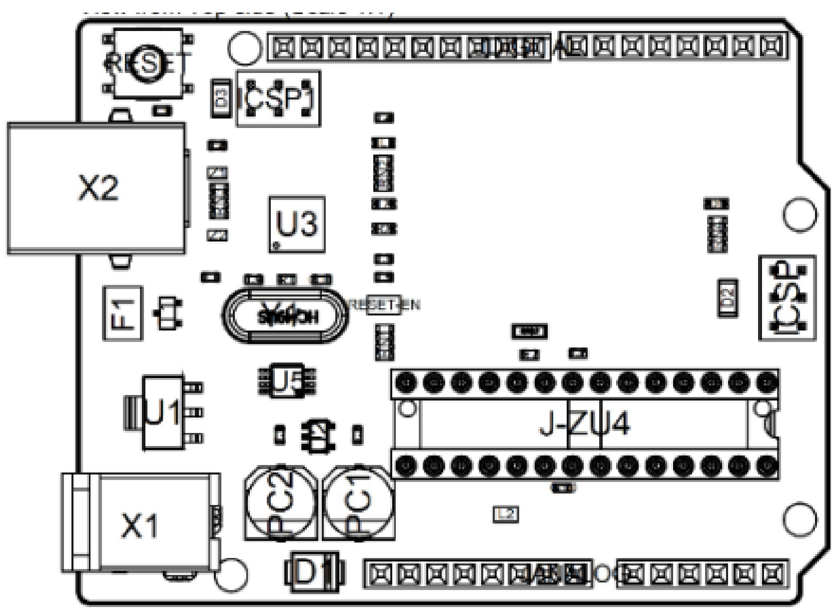
- Watchdog Timer with separate on-chip oscillator
- Six PWM channels
- Interrupt and wake-up on pin change

- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - debugWIRE interface for on-chip debugging and programming
- **Power**

  2.7-5.5 volts

Board Topology

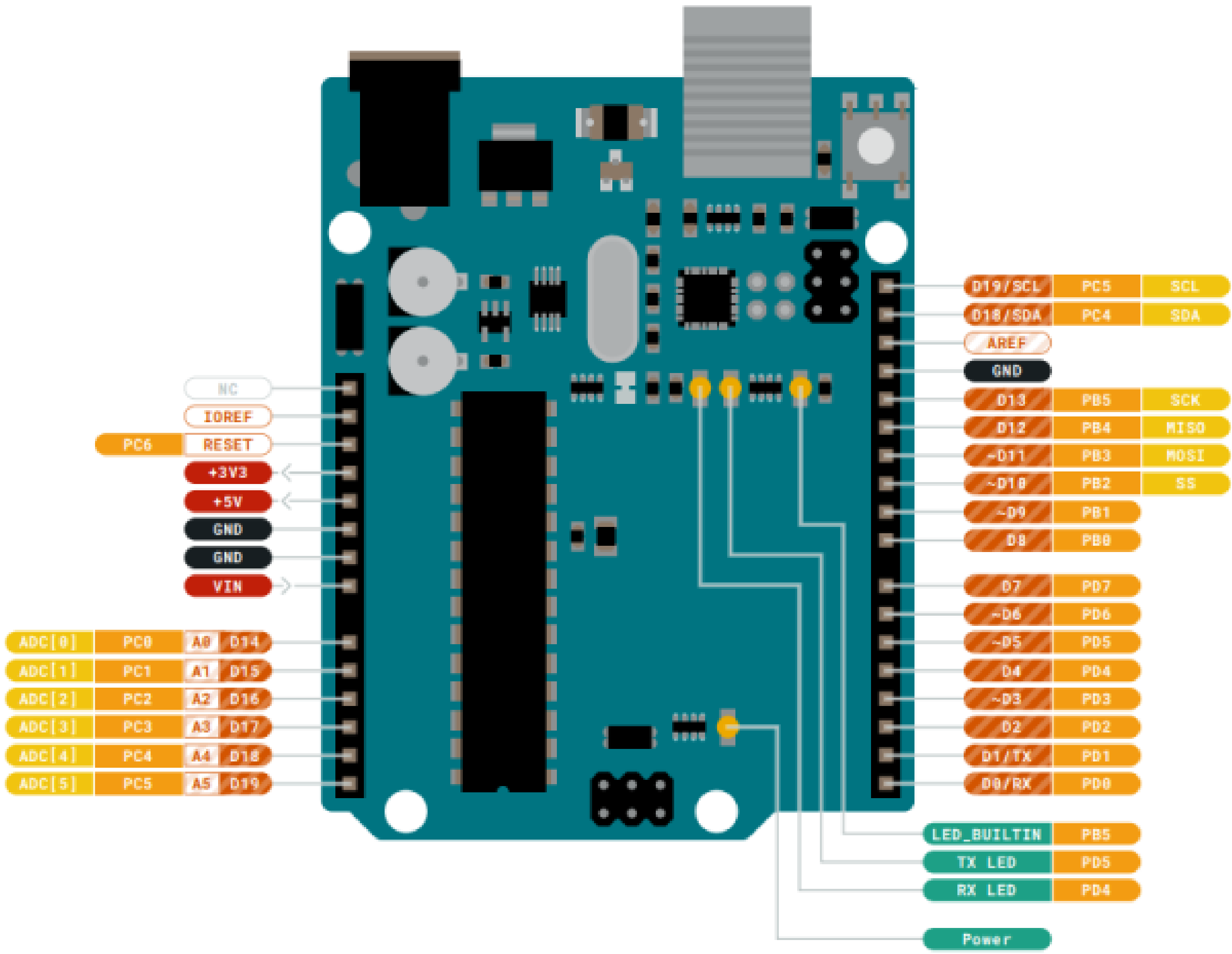
Top view



Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

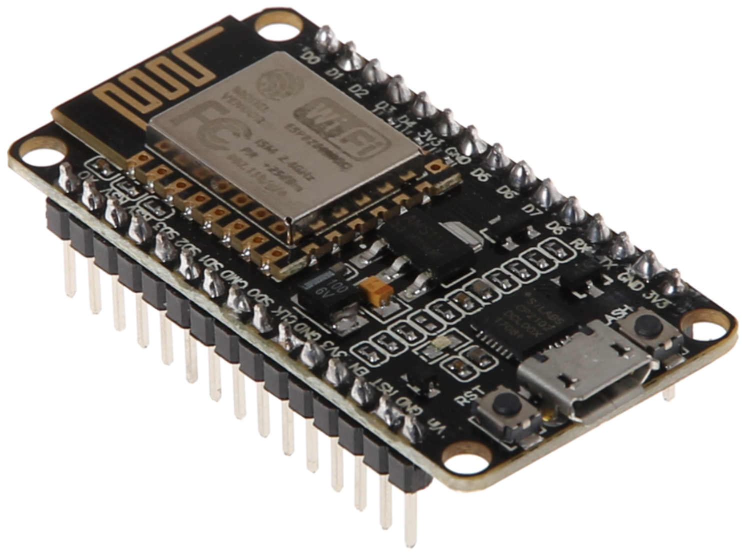
Connector Pinouts



NC
IOREF
PC6
RESET
+3V3
+5V
GND
GND
VIN
ADC[0] PC0 A0 D14
ADC[1] PC1 A1 D15
ADC[2] PC2 A2 D16
ADC[3] PC3 A3 D17
ADC[4] PC4 A4 D18
ADC[5] PC5 A5 D19
D19/SCL PC5 SCL
D18/SDA PC4 SDA
AREF
GND
D13 PB5 SCK
D12 PB4 MISO
~D11 PB3 MOSI
~D10 PB2 SS
~D9 PB1
D8 PB0
D7 PD7
~D6 PD6
~D5 PD5
D4 PD4
~D3 PD3
D2 PD2
D1/TX PD1
D0/RX PD0
LED_BUILTIN PB5
TX LED PD5
RX LED PD4
Power

Pinout

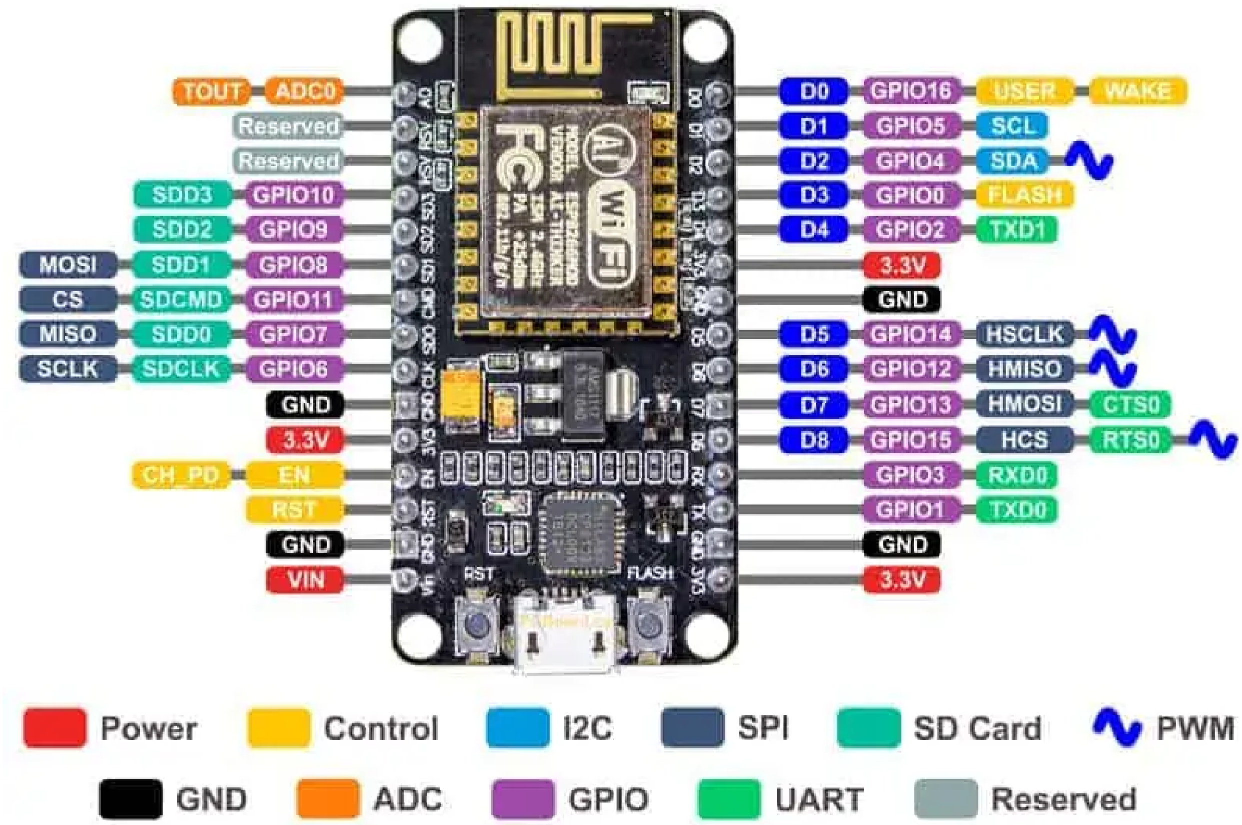
nodeMCU ESP8266



NodeMCU ESP8266 Specifications & Features

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

Board Topology and Connector Pinouts



Power Pins There are four power pins. VIN pin and three 3.3V pins.

- VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the VIN pin
- 3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components.

GND are the ground pins of NodeMCU/ESP8266

I2C Pins are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

SDIO Pins NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs (100 Hz and 1 kHz).

Control Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- EN: The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- RST: RST pin is used to reset the ESP8266 chip.
- WAKE: Wake pin is used to wake the chip from deep-sleep.

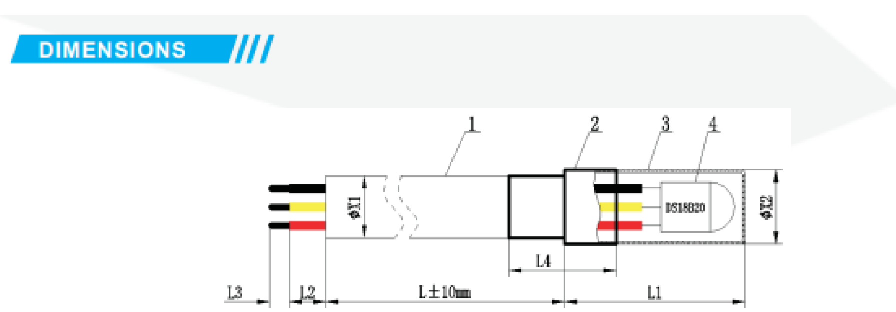
Control Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- EN: The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- RST: RST pin is used to reset the ESP8266 chip.
- WAKE: Wake pin is used to wake the chip from deep-sleep.

Temperature Sensor DS18B20 (GTS200)

Temperature Sensor DS18B20 (GTS200) Specifications & Features

- Arduino Uno V3 Specifications & Features
- Power supply range: 3.0V to 5.5V
- Operating temperature range: -55 ° C to + 125 ° C (-67 ° F to + 257 ° F)
- Storage temperature range: -55°CC to + 125°C (-67F to + 257F)
- Accuracy in the range of -10 ° C to + 85 ° C: ± 0.5 ° C
- Waterproof stainless steel sheath
- Sheath size: 6 * 50mm or custom



1	Cable length, wire diameter OD	PVC cable*3P	OD5.0mm
2	Heat shrinkable tube size	Heat shrinkable tube	OD6*30mm
3	Probe size	Stainless steel probe	OD6*30mm
4	DS18B20 sensor		Red VDD is the external power supply input The yellow DQ is a digital signal input / output terminal

PARAMETERS

Digital chip	1DS18B20
Probe size	Φ 8mm, φ6mm (inside), length = 30mm or custom
Insulation Materials	Glass fiber, PVC, Teflon, silicone rubber or custom made
Wire material	FRP, PVC, Teflon, silicone rubber,stainless steel braided
Shell material	Stainless steel · nickel plated copper, brass, plastic
Wire Connector	UL Series(such as UL1007),Supply wire number, using temperature range, outside diameter and material requirements.
Special requirements	Molex, JST, DuPont, CWB, CJT , SM , TJC3, PH, EH, 5264, U-type etc.
Wiring	Waterproof, acid proof, antiseptic
	Black: GND Yellow: DATA Red: VDD+

ORDER INSTRUCTION

When you placing order , please inform us the following parameters:

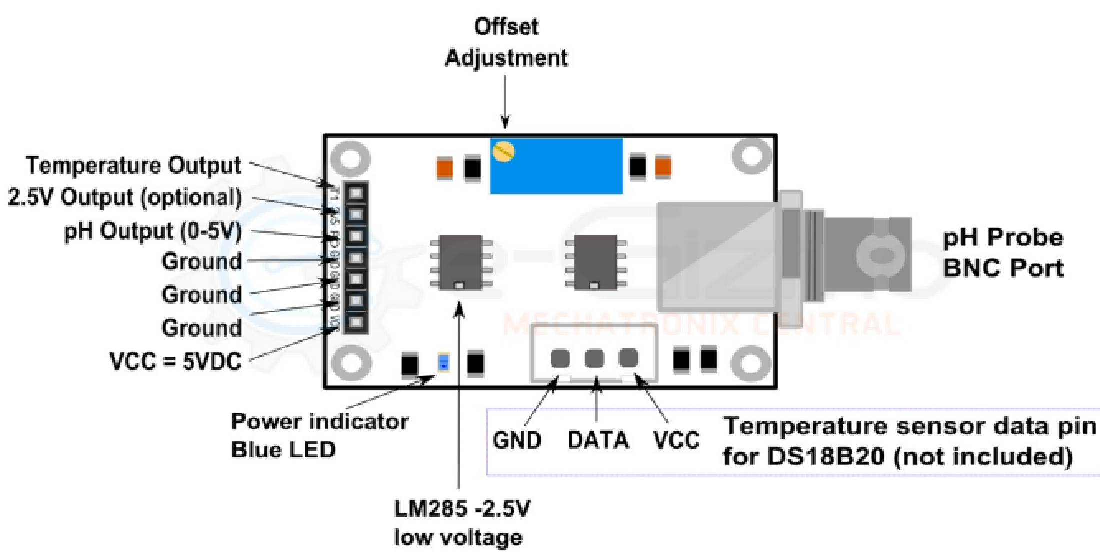
1. Application and working environment (whether to be waterproof, acid or alkali and others)
2. Shell diameter D and length L (commonly size: 6 * 30mm, 6 * 50mm, other can be customized)?
3. How many lines of output? (Commonly used 2-wire/ 3-wire, choose one) ?
4. Wire material and length (commonly wire PVC, Silicone, Teflon wire)?
5. Temperature range (temperature range cannot exceed -50 to +125 °C)?
6. How to deal with the cable end (hanging tin or with connector)?

pH Sensor E-201-C

pH Sensor E-201-C Specifications & Features

- **Input supply voltage:** 5V
- **Working current:** 5 - 10mA
- **Detection concentration range:** PH 0 -14
- **Detection range of temperature:** 0 - 80 degC
- **Response Time:** ≤ 5S
- **Stability Time:** ≤ 60S
- **Output:** Analog
- **Power Consumption:** ≤ 0.5W
- **Working Temperature**: -10 to +50 deg C
- **Working Humidity:** 95% RH (nominal humidity 65% RH)
- **Weight:** 25g
- **PCB Dimension:** 42mm x 32mm x 20mm

pH Sensor module kit v2 pinouts



IMPORTANT TIPS:

On How to Clean pH Electrodes:

1. Do not "wipe" or rub the electrode.
2. Swirl the electrode gently in the cleaning soultion.
3. Gently rinse with deionized or distilled water.
4. Store in a storage solution.
5. When possible, use a specialized electrode.

Which Cleaning Solution to use?

The cleaning solution you use will depend upon your particular process and the residues you are trying to remove. There is a wide range of pre-mixed cleaning solutions available online or you can make your own. Make sure you take care when handling any cleaning solution – some can be hazardous so make sure you follow all safety instructions and wear appropriate protection equipment!

(For more information - Please go to reference link)

Reference: https://www.southforkinst.com/ph-electrodes-clean-repair/

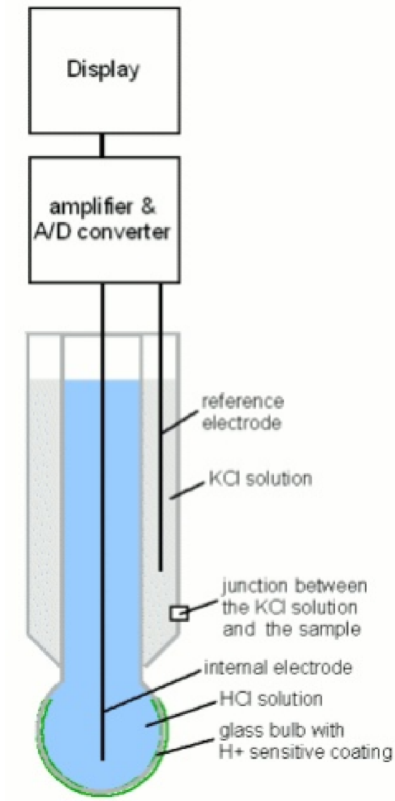


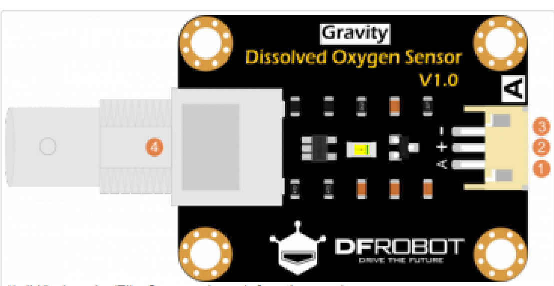
Figure 3: pH Electrode parts

Gravity: Analog Dissolved Oxygen Sensor SKU:SEN0237

Gravity: Analog Dissolved Oxygen Sensor Specifications & Features

- **Dissolved Oxygen Probe**
  - **Type:** Galvanic Probe
  - **Detection Range:** 0~20 mg/L
  - **Temperature Range:** 0~40 ℃
  - **Response Time:** Up to 98% full response, within 90 seconds (25℃)
  - **Pressure Range:** 0~50
  - **PSI Electrode Service Life:** 1 year (normal use)
  - **Maintenance Period:**
    - ➔ **Membrane Cap Replacement Period:**
      - 1~2 months (in muddy water);
      - 4~5 months (in clean water)
    - ➔ **Filling Solution Replacement Period:** Once every month
  - **Cable Length:** 2 meters
  - **Probe Connector:** BNC
- **Signal Converter Board**
  - **Supply Voltage:** 3.3~5.5V
  - **Output Signal:** 0~3.0V
  - **Cable Connector:** BNC
  - **Signal Connector:** Gravity Analog Interface (PH2.0-3P)
  - **Dimension:** 42mm * 32mm/1.65 * 1.26 inches

Module Overview



(/wiki/index.php/File:Oxygen_board_function.png)
Dissolved Oxgen Sensor

No.	Label	Description
1	A	Analog Signal Output (0~3.0V)
2	+	VCC (3.3~5.5V)
3	-	GND
4	BNC	Probe Cable Connector

Tips in using the Sensor

- Before using the dissolved oxygen probe, 0.5 mol/L NaOH solution should be added into the membrane cap as the filling solution of the probe. As NaOH solution has strong corrosivity, protective gloves should be put on before handling the solution. If the solution accidentally drops onto the skin, wash your skin with plenty of water immediately.

- The oxygen permeable membrane in the membrane cap is sensitive and vulnerable. Be cautious when handling it. Fingernail and other sharp objects should be avoided.

- During the measuring process, the oxygen probe will consume a little oxygen. You need to gently stir the water and let the oxygen be distributed evenly in water. On the other hand, do not stir violently to prevent the oxygen in the air from quickly entering into the water.

Requirements

- **Hardware**
  - Dissolved Oxygen Probe (With Membrane Cap) x 1
  - 0.5mol/L NaOH Solution x 1
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - Dissolved Oxygen Signal Converter Board x 1
  - Analog Cable (3Pin) x 1
- **Software**
  - Arduino IDE (Version requirements: V1.0.x, V1.6.x or V1.8.x), Click to Download Arduino IDE from Arduino® (https://www.arduino.cc/en/Main/Software%7C)

JXBS-3001-NPK-RS Soil NPK Sensor

JXBS-3001-NPK-RS Soil NPK Sensor Specifications & Features

Parameters	Technical Specs
Measure Range	0-1999mg/kg
Accuracy	±2%F.s
Resolution	1mg/kg(mg/l)
Response Time (T90, Seconds)	<10s
Working Temperature	5-45℃
Working Humidity	5-95%RH (Relative humidity), no condensation
Baud Rate	2400/4800/9600
Communication Port	RS485
Power Supply	12V-24V DC

Interface Description

The power interface can be 12-24V for wide voltage power supply. When wiringthe485 signal line, note that the A/B lines cannot be connected in reverse, andtheaddresses between multiple devices on the bus cannot conflict.

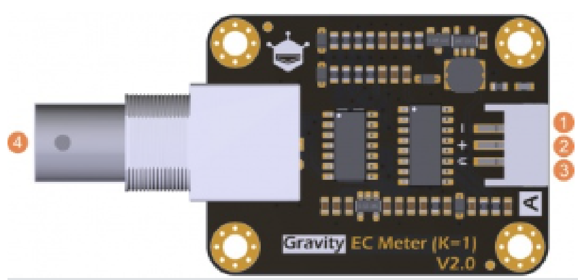
Function	Cable Color	Specs
Power	Brown	Power supply +
	Black	Power supply -
Communication	Yellow (grey)	485-A
	Blue	485-B

Gravity: Analog Electrical Conductivity Sensor / Meter (K=10) SKU:DFR0300-H

JXBS-3001-NPK-RS Soil NPK Sensor Specifications & Features

- **Signal Conversion Board (Transmitter) V2**
  - **Supply Voltage:** 3.0~5.0V
  - **Output Voltage:** 0~3.4V
  - **Probe Connector:** BNC
  - **Signal Connector:** PH2.0-3Pin
  - **Measurement Accuracy:** ±5% F.S.
  - **Board size:** 42mm*32mm/1.65in*1.26in
- **Electrical Conductivity Probe**
  - **Probe Type:** Laboratory Grade
  - **Cell Constant:** 1.0
  - **Support Detection Range:** 0~20ms/cm
  - **Recommended Detection Range:** 1~15ms/cm
  - **Temperature Range:** 0~40°C
  - **Probe Life:** >0.5 year (depending on frequency of use)
  - **Cable Length:** 100cm

Board Topology



Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.0~5.0V)
3	A	Analog Signal Output(0~3.4V)
4	BNC	Probe Connector

APPENDIX D

Bill of Materials

	Component	Description	Price (sum)
Power System	Power Bank	ROMOSS Sense6P 20000mah	₱848.00
	Battery	Panasonic Sealed Rechargable Battery 12V	₱850.00
	Pocket Wifi	Smart Bro w/ micro usb cord	₱600.00
	Solar Panel	CL-1615 16V 15W Solar Panel	₱518.00
Microcontroller	Arduino	Arduino Uno	₱750.00
	nodeMCU	8266 v3	₱160.00
Sensors/Module	Temperature Sensor	DS18B20 Temperature Sensor w/ module	₱90.00
	pH Sensor	E-201-C pH Probe + Ampl	₱934.00
	Dissolved Oxygen Sensor	Gravity: Analog Dissolved Oxygen Sensor w/ module	₱12,000.00
	NPK Sensor	Soil Nutrient Intelligent Detector Tester Meter NPK Sensor	₱3,100.00
	Conductivity Sensor	Analog Electrical Conductivity Probe w/ module	₱4,749.00
Buoy	Plywood	1/4" thickness	₱1,000.00
	Floater	Heavy Duty Fiber Ring	₱1,000.00
	Insulation Foam	1m length	₱80.00
	Rope	4m length	₱50.00
	Marine Epoxy	2pcs.	₱500.00
	Primer	Flatwall Enamel	₱180.00
	Hook	4pcs.	₱20.00
	Paint Thinner	1 bottle	₱50.00
	Yellow Paint	Quick Dry Enamel	₱280.00
	Tumblr	Plastic Tumblr	₱50.00
	Container	Transparent	₱7.00
Materials	Cutter Blades	1 set	₱10.00
	Glue Stick	Small size 20 pcs.	₱80.00
	Sandpaper	#1000 2 pcs. & #120 2 pcs.	₱60.00
	Super Glue	Mighty bond 1g 1pc. + 10g 4pcs. + Dolphband 12pcs	₱626.00
	Water Proof Hose	Cambric Tubing 12MM	₱69.00
	Jumper Wires	10cm 1 pack + 30cm 1 pack + 20cm 10 pcs.	₱290.00
	Electrical Tape	1pc	₱50.00
	Solid Wire	5m (red and white)	₱30.00
	Common Nail	1/4 kilo	₱30.00
	Baby Roller	1pc.	₱70.00
	PCB	PCB hole plate glass fiber epoxy plate board	₱100.00
	Paleta	2pcs.	₱60.00
	Pakong Bakya	1 set	₱15.00
	Card Reader	1pc.	₱100.00
	Ferric Chloride	1 Bottle	₱100.00
	Wall Putty	2 pcs. Bosny 1kg	₱299.00
	Cambric Tubing	1pc	₱209.00
	Soldering Lead	1 Roll	₱150.00
		TOTAL:	**₱30,134.00**

APPENDIX E

Documentation


P&G


SRIXON

APPENDIX F

User's Manual