Student Name: M. Vihanga Deshan Peiris

Student Number: 2057112

Module Code: 6CS007/CN4

Module Name: Project and Professionalism

Project Title: IoT based Pet Fish Farming Management System

Module Leader Name: Mr. Dhishan Dhammearatchi

Supervisor Name: Mrs. Dharani Abeysinghe

Reader Name: Mrs. Vigini Mekala

Submission Date: 23rd January 2022
(The date you handed in the assessment)

Award Title**:** Computer Networks
(Award Title for your project, if in doubt refer to your course/Module Registration)

School of Mathematics and Computer Science

Declaration Sheet

This document is being submitted in partial fulfilment of the assessment requirements for the aforementioned award.

This work, in whole or in part, has not been previously presented to the university or to any other institutional body, for assessment or for any other reason. With the exception of any explicit acknowledgements, references, and/or bibliographies listed within the work. I attest that the intellectual content of the work is entirely the result of my own efforts and not those of anyone else.

Copyright is stated to belong to the author of any project effort. By submitting such copyright work for evaluation, the author provides the University a lifelong royalty-free license to do all or any of the acts specified in section 16(i) of the Copyright Designs and Patents Act 1988. (For example, to duplicate the work; to distribute copies to the public; to perform, display, or play the work in public; to broadcast the work; or to adapt the work.)

Student Name (Print): M. Vihanga Deshan Peiris

Student Number: 2057112

Signature: ………………………
Date: 23/01/2022
(Must include the unedited statement above. Sign and date)

Please use an electronic signature (scan and insert)

Abstract

The suggested function is an embedded system for controlling aquaculture automatically. Aquaculture, sometimes called aquafarming, is the practice of cultivating aquatic organisms such as fish, crustaceans, and crabs while limiting dangers through the use of various sensors. The suggested method enables remote monitoring and management of a fish farming system using the Internet of Things (IoT). The objective of this article is to design an automated monitoring system for fish farming that will save farmers time, money, and energy.

The Internet of Things has revolutionized the country's agricultural industry. We use a variety of sensors in the fish farming process, including pH, temperature, and level sensors. All work can be automated with these sensors, and the fish farming process may be watched remotely from other locations.

Acknowledgement

I would want to use this opportunity to express my gratitude to everyone who assisted me in completing this project. And I want to express my heartfelt gratitude to my supervisor professor, Ms. Dharani Abeysinghe, for assisting and encouraging me during this work. Additionally, I am really appreciative of the assistance provided by my batch mates, family members, and friends in assisting me in completing my project. Finally, I'd want to express my gratitude to all of the well-wishers named and unmentioned who assisted me in completing this task.

Table of Contents

List of figures

Chapter 01

1.1 Background

At present there are many small-scale fish farmers in Sri Lanka who engage in fish farming while conducting other part-time jobs or other secondary businesses. Therefore, it is beneficial for them to have a system where they can monitor their fish farms remotely since it is crucial for them to monitor the condition of the water in fish tanks continuously since any minor changes in the water condition can affect the fish. The system which will be proposed in this proposal will provide visibility for the fish farmers via a web dashboard to monitor the key factors in fish farming which are temperature, pH level and the water level of the tanks from any location in the world.

In this system the temperature of the water will be monitored constantly and if the temperature decreases than the normal level, the water heater will be activated automatically to control the temperature of the water. Similarly, the pH level of the water will be monitored in a continuous manner. Generally, the pH level of the water does not change unless the water gets contaminated with other chemicals or if the water gets stagnated. Through this system the fish farmers are able to pre-plan the schedule of changing the water in the fish tanks automatically without any human interventions. This system will assist the fish farmers to continuously assess the condition of the water in the fish tanks and it will also save time and increase efficiency of the fish farm by automating the data collection of the condition of the water and by automating the temperature control of the water and replenishing the water in fish tanks.

Another important factor considered in fish farming is maintaining the accurate levels of fish population in each tank because there should be sufficient level of space available for the fish to grow. Since the fry are transported in large bags it is difficult to measure manually how many fry is available in one bag. This system calculates the fish population of each tank through image processing using multiple pictures and if the number of fish exceeds then fish farmer can reduce the number of fish,

Today there are many small-scale fish farmers in Sri Lanka. Most of them are engaged in such activities while doing other work and this system is very useful for such people to save their time. This system makes it very easy for them to manage the water in the tanks. And they can look at a web dashboard from anywhere to see if system have managed the water properly. Other key factors in fish farming are temperature, pH, and the water level in the tank, which can be observed from anywhere. If these fish tanks are not properly monitored, it can affect every fish in them. Here the pH is monitored from time to time so that can know if there is any change in the water that is harmful to the fish. Similarly, the temperature of the tank is monitored from time to time so that can determine if it fits the type of fish in the tank. If there is any change in temperature, the heater will automatically activate and control it. Normal pH changes do not always occur. It changes when any of our contaminants are added to the tanks, or the water becomes stale. This system is designed to facilitate the day-to-day work of fish farmers so here they can also change the water according to a pre-planned schedule. So, farmers can save a lot of time. Another factor when considering pet fish farming is the fish population in each tank. Because the bags carry a lot of fry at one time, the tank does not have enough space as the fry get bigger. The solution is to calculate the population and determine how many fish are in one tank. If the number of fish in the tank is exceeded, the farmer can know about it. This is done by image processing multiple pictures.

1.2Aims of the Project

The aim of this project is to provide a solution for fish farmers where they are able to monitor key factors affecting in fish farming which are temperature, pH level and the water level in the tanks, from any location in the world.

- Generally, fish farmers do their day-to-day activities in the farm manually.
- Due to the above reason the unnecessary amount of time is consumed for simple tasks and sometimes due to errors in observations financial losses can be occurred.
- When using this system, they can do conduct their activities with minimum risk and also save time as well.

1.3. Academic Questions and Objective

1.3.1. Academic Questions

How to properly inspect fishing tanks and see if there are any changes in the water that are harmful to all the fish?

- How to measure the PH level of the water is good for fish?
- How to remove drainage and refill the tank with fresh water?
- How can feed to the fish according to the pre-planned time?
- How to inform the farmer via SMS if the environmental conditions in the tank change adversely?
- How to calculate the fish population in the tank and maintain the tank under proper conditions?

1.3.2 Objectives

The objectives of the IoT based Pet Fish Farming Management System

- Automate wastewater drainage and replenishment of water in fish tanks when pH level of the tank increases.
- Automate the feeding schedule to a pre-scheduled time,
- Alert the farmer via SMS if the water condition of the fish tank changes.

1.4 Scope

At present many small-scale fish farmers operating in Sri Lanka are engaged in part time jobs and other secondary businesses. Therefore, this management system is useful for them to continuously monitor the key factors affecting the water condition of the fish tanks such as temperature, pH level and water level from any location. This system will assist the farmers to stay updated in the conditions of the fish tanks and it will also help the farmer to save time by automating activities such as water replenishment of the tanks and feeding.

Today there are many small-scale fish farmers in Sri Lanka. Most of them are engaged in such activities while doing other work and this system is very useful for such people to save their time. This system makes it very easy for them to manage the water in the tanks and monitor the key factors in fish farming which are temperature, pH, and the water level in the tank, which can be observed from anywhere. Here the pH is monitored from time to time so that we can know if there is any change in the water that is harmful to the fish. So, farmers can save a lot of time.

1.5 Structure of the Report

The structure of the report will be as follows,

- Chapter 01 of the report will provide an introduction to the project with aims and objectives of the project.
- Chapter 02 will be consisted of the literature review where findings of scholarly articles related to this project will be provided.
- Chapter 03 will explain the design and structure of the research and what techniques was used in collecting the data.
- Chapter 04 will discuss regarding the project’s artifacts.
- In chapter 05 the conclusion derived from the project will be discussed.
- Lastly, references will be provided to support all the information included in this proposal.

In Chapter 02 of the Literary Review, this report is organized as follows. Introduces the project and others like it. In Chapter 3, project methodology is also explored. The project's artifact is also discussed in Chapter 4. The conclusion is discussed in the chapter 05. Finally, the points have been supported with references.

Chapter 02

2.1 Introduction

IoT based Pet Fish Farming Management System facilitate to monitor the key factors affecting the water condition in fish farming such as temperature, pH level and water level from any location in the world. It is crucial to maintain the required water condition in the fish tanks because it can severely affect the wellbeing of the fish in the tank. Generally, the pH level of the water tank does not change unless there the water gets contaminated with other chemicals or if the water gets stagnated. Therefore, this system is designed to facilitate to support day-to-day activities of fish farmers which also includes automating the replenishment of water levels to a pre-planned schedule. Similar projects under following topics will be analysed under literature review,

- Fish-tank Management System
- Web base Realtime Fish-tank Monitoring
- Image-based Population Calculation

2.2 Similar Projects

Fish-tank Management System

Aquaculture research as a method for increasing stable production. The project's main aim is to reduce dangers by remotely monitoring the fish farming system using various sensors. Sensors such as pH, temperature, and level sensors are used in these operations. All the labor is automated with these sensors, and it will be simple to monitor the fish farming remote from another location. Aquaculture research as a method for increasing and stabilizing productivity. Fish are cold-blooded animals that regulate their body temperature by interacting with the water. The amount of dissolved oxygen in the water and fish oxygen intake are affected by changes in water temperature. Ignoring the fact that the fish can withstand a great variety of water temperatures, any rapid, extreme changes in water temperature will have a significant impact on the fish's physiology. A chilling injury will cause the fish to become paralyzed and lose their balance, resulting in death. The respiratory center could be to blame, or water control could be disrupted by high temperatures. The fish have respiratory arrest when the water temperature rises. The amount of dissolved oxygen in water changes with the seasons, according to Fish World magazine. Fish physiological rate increases when water temperature rises, resulting in less dissolved oxygen in the water. Low water temperature slows fish physiology and raises dissolved oxygen levels in the water. Fish growth will be hampered if the amount of dissolved oxygen in the water falls below a certain threshold. Fish will die if the amount of dissolved oxygen falls below the level necessary to live. In general, the acidity and alkalinity of a solution in fish farming should be kept between 6 and 8. Acid erosion of the gill tissue, tissue disintegration necrosis, excessive mucus secretion, stomach congestion, and inflammation are all side consequences of being too acidic or alkaline. The fish will die if the PH value is less than 4.5. (RAJA Subramanian, 2017)

Agricultural productivity in the country has been changed thanks to Internet of Things (IoT) technologies. As a result, this study proposes a concept for recognizing remote monitoring of the fish farming system utilizing various sensors in place to evade hazards. Use a variety of sensors in this study, such as pH, temperature, and level sensors. All the work is automated with these sensors, and it will be simple to monitor the fish farming remotely from another location. furthermore, in recent years, the Internet of Things has successfully proved its wide range of applications. However, today's fish farms are infrequently equipped with advanced devices that provide real-time and related water monitoring capabilities. There are countless examples of how the Internet of Things could help aqua culturists improve their working conditions. Some fish ranch, for example, are located far from the land, and using IoT to screen water at a distance could help them save money. Another concept is that changes in water quality can happen at any time and in any way, therefore always monitoring water with cautions cannot miss a specific event. Fisheries and aquaculture is the main source of income of the 10-12 percent of the global population, which is around 660-820 million people. Regarding sexual orientation balance, neediness, and sustenance security, the portion has a critical role to play. With a global fish supply of more than 150 million tons, more than 85 percent of this supply is used solely for food, providing 15 percent of the world's protein and basic nourishment for over 4.3 billion people. (Janet, Balakrishnan and Sheeba Rani, 2019)

Data is collected from ponds in various locations to observe the standard water quality parameters which are acceptable in fish farming and the causes of diseases occurred in fish due to changes in the water quality parameters. To ensure the accuracy of the data collected water level, mistake rate and multiple machine learning techniques were used to conduct this study. The trained and test parts have been better fitted by logistic regression. To ensure if the condition of the water is suitable for fish farming according to the quality standards different calculations were conducted. An experiential IoT-based system architecture has been used to contrast the forecast of the study. In addition, this study investigated the viable environmental factors required for development of the fish and causes and threats of fish death and growth rate of the fish. To monitor the standards of the water quality it is suitable to develop an affordable and convenient fish farming system. Determinants of the water quality such as pH level, total dissolved solids, nitrite, nitrates, temperature of the water, alkalinity, ammonia, total suspended solids, dissolved oxygen, hardness, biochemical oxygen demand, and others will be measured using IoT based sensors. In addition, in this affordable system uses a high number of sensors to analyse the values of the water quality parameters. Also, machine learning systems are used to evaluate how standard water condition can be achieved in developing a fish farm with the highest accuracy. To analyse further, the data collected through these systems are stored in a centralized database in a cloud server. These data can be used to identify pregnant fish from their behavioural patterns, food consumption patterns and to predict the suitability of the quality of the water for fish farming. (Ahmed, Rahaman, Rahman and Abul Kashem, 2019)

A supplier of clean water, a way to discharge wastewater, and effective water control systems are all infrastructure requirements for establishing catfish farms. The operators of these fishponds confront a few obstacles, including the need to feed the fish on a regular basis, monitor the water quality, and change the water when the condition becomes harmful for the fish. An Internet of Things based approach is used for automating farm management and allowing remote pond monitoring and management. The system consists of a pond controller that monitors the pond's water quality using appropriate sensors. The actions around the pond are recorded by a CCTV system and stored in the cloud. The Pond Controller oversees the fish's regular feeding system as well as the pond's water control system. The system is also developed to be operated remotely using a specially created mobile application that accesses the CCTV files and regulates the pond controller's operation. This system will allow the operation of one or more fishponds from a single mobile device, lowering management expenses and improving the quality of the fish farms' produce. (Idachaba, Olowoleni, Ibhaze and Oni, 2017)

As a consequence of continuous growth of the scale and concentration of the aquaculture industry, latest aquaculture technologies have been forced to excessive production which have led disproportions in the water environment, development of fish diseases, and decrease in the quality of the aquatic products at a rapid rate. In addition, due to increase in the average age of the agricultural worker in many areas of the world, labour shortage will be experienced in seafood production industry and practices in aquaculture industry will be rapidly changed. As modern information technology has continuously entered number of agriculture fields, the concepts of a smart fish farm has entered the industry. The fish farm looks for ways to optimize all functions related to fish farming such as managing the oxygen levels in the water, improve feeding systems, minimize incidence of diseases, and accurate harvesting by replacing the workforce with machines to completely free from human intervention and achieve a green and sustainable aquaculture business. This reports the use of factory intelligent equipment, IoT, edge computing, 5G, and intelligent systems in modern aquaculture, also the restrictions and opportunities that have emerged. Meantime develops frameworks for key functional modules in developing a fish farm is offered based on the different requirements of the business. (Wang et al., 2021)

Realtime Fish-tank Monitoring

Today, guppy fish farming is still monitored manually. To make things easier for guppy fish growers, a monitoring and control system is required. A sensor module plus a monitoring module makes up this system. The pH and saltiness of the water are detected by the sensor module and sent to the monitoring module. Monitoring is done using IoT technologies and a web-based system. The pH value is monitored and controlled by this web-based system, which simply monitors the salinity value. The monitoring and control system gather all the data regarding the pH level and salinity shared by the sensors, this information will be documented and stored in a database. Wireless media will be utilized to display data and information from the database on the website, which will assist people to analyse the data information from any location. Additionally, farmers will be able to analyse the pH and saline level in the water. If the pH level is below 6.5 and above 7.5, the fatality of the salt concentration level in the water is low. If the pH level is below 160 and above 210, the fatality of the salt concentration level in the water is high.

Additionally, the farmers can also use the website to change the pH level with the servo motors, granting them to identify whether the pH level contained in the water is dangerous to guppy fish. (Periyadi, Gita Indah Hapsari, Zahir Wakid, Sobran Mudopar, 2020)

Moreover, large number of sensor nodes are used in the system design to calculate the value of the water quality parameters, with the system focused on the limited system. Additionally, a computer vision system is used to examine the suitability of the water condition to develop the fish farm to achieve high level accuracy. The data gathered will be saved in a centralized database in a cloud server to conduct further researches such as identifying pregnant fish according to their behavioural changes, food consumption patterns and forecasting the water condition of the water which is suitable for fish farming. Arduino is in responsible of sensor acquisition, and the Raspberry Pi serves as both a data processor and a server. Raspberry Pi also takes photos with the help of the smartphone to detect the color of the water. The terminal device is an Android phone. An android app can be used to monitor the water quality using Wi-Fi within the Wi-Fi range and the Internet from anywhere in the world. The four parameter values are analyzed to determine the overall estimated state of the water and the necessary action. Every function of this checking device is reliable and simple to use. (Sajal Saha *et al*….,2018)

This paper describes a development of a functional Internet of Things (IoT) system to monitor fish tanks and also the IoT system consists of a small board computer that examines the data and provide visual and audio alerts to the fish farming manager, as well as various sensors that identify important factors of water quality such as temperature, light intensity, or water level. The present system is unable to analyse the data and share it to the final users via a website or mobile application. Since most of the fish ponds are located at remote areas because they rely on fresh water, they can use an expansion module like as the Used that modem allow the final user to observe and control relevant sections of the fish farming pond through an IoT system at real time. All-in-one Wivity modem allows users to communicate with IoT systems via WiFi, LTE, LoRaWAN, or satellite connectivity. This module can relate to IoT platforms like as Japser, Microsoft Azure, or Amazon Web Services in the future. For future development, we aim to expand not just the appropriate the services on other platforms, but also the existing IoT system for specific fish species by adding more control modules an sensors. (Nocheski S. and Naumoski A, 2022)

Fish had been left due to a lack of maintenance, such as dirty aquarium water or fish spawning ponds. One of the solutions to the difficulties is an IoT-based smart aquarium monitoring system. This study demonstrates a sample of an Internet of Things-based Smart Aquarium Monitoring System for maintaining fresh water in aquariums for fish habitats. The system's purpose is to keep an eye on the fresh water to provide a healthy fish habitat. This technology functions as a fish feeding system that is operated using a smartphone. The designed system makes use of Arduino MEGA and NodeMCU controllers. Wi-Fi communication on the NodeMCU is utilized to govern the operation between the smartphone and the controller. The pH value of water is measured using an analog pH sensor and displayed using a Liquid Crystal Display (LCD). The Arduino Software IDE is used to develop the coding, while the BLYNK software is utilized to create Android software applications. The system is designed to observe the pH value that is suitable for the variety of fish and to manage fish feeding via an Android application on a smartphone. This research is important for the development of the IR4.0 system in supporting fish pets, and a larger project for fish breeding in the pond can be sampled with this project, which has economic implications for the country. (Pasha Mohd Daud, Sulaiman, Mohamad Yusof and Kassim, 2020)

The main aim and goal of this study is to measure water quality parameters that are suitable for a variety of aquaculture living species, particularly fish. Five sensors are placed in a one-tone fish tank to calculate and observe the oscillations in water standards, particularly during feeding time. A waterproof temperature sensor, a water PH sensor, a water turbidity sensor, an air temperature sensor, and a light sensor are among the smart sensors. These sensors are attached to an Arduino board, which feeds the sensor data to the GSM and then to the Thing Talk cloud, which is a simple way to observe data changes 24 hours a day. As a result, the information gathered of water parameters appears to fluctuate slightly, although this has no effect on the health of the fish in the tank. The purpose of using these sensors is to show the statistical relationship between air temperature and water temperature, light intensity and turbidity, and turbidity and PH, particularly during feeding times. These IoT sensors are inexpensive to purchase and install, as well as straightforward to monitor on the cloud. As a result, lot of farmers who want to sustain their aquacultural species safe and reproducing may find this to be highly cost-effective. (Nasir and Mumtazah, 2020)

The goal of this study is to automate the management and maintenance of cold water in aquariums. To construct a water management system for a tank, the researchers combined various sensors. The water level sensor was one of the sensors used by the researchers to determine if the water level in the aquarium being filled by the water pump is already at the correct level, in which case it will stop filling, and if not, it will continue to fill the aquarium. The researchers also employed a turbidity sensor that will be attached to the water pump. If the turbidity sensor detects that the water quality is decreasing, the water pump will open, and the water will be pumped out. The temperature of the water is another sensor. To achieve the research's goals, all the sensors are connected to a microcontroller. The researchers used the PHP server to allow the user to observe data from all the microcontroller's sensors. The information will be kept in a table and sent to the PHP server, where it will be used for data login. The data that may be viewed on the server is updated every hour. The goal of this study is to automate the management and maintenance of cold water in an aquarium. (Cruz, Fulla, Sorezo and Balba, 2022)

Image-base Population Calculation

WSP's Ichtyo System is the result of ten years of development and study. It allows fish to be monitored remotely and in real time as they pass through the system, minimizing the stress of routinely capturing individual fish. Furthermore, independent of local conditions, it integrates numerous technologies to precisely determine essential biological factors. Complementary technologies allow for the circumvention of individual limits, accurate estimations, and long-term monitoring of fish populations. Every river is unique, which is why our diverse team of experts customizes a solution for each project. We're continually coming up with new techniques to speed up data processing and increase the system's reliability. Humans can better manage a larger number of sites to protect migratory fish species by lowering monitoring costs. IchtyoSTM is a one-of-a-kind system that combines multiple forms of technologies to enable more exact estimates and long-term monitoring of fish populations. It is made up of three light curtains that are formed by three detecting bands. Various sensor bands are engaged when the fish pass through the laser beams, and a complicated algorithm analyses the activation patterns to classify and measure the individual fish. A camera records each fish that goes by at the same time. Our online portal then publishes the recorded video as well as the counting parameters. (Automated Counter for Fish Populations Monitoring, 2022)

Large, farmed fish, such as tuna, are frequently counted and sized during their transit from one net cage to another. DIDSON (dual-frequency identification sonar) is a fish counting and size tool that is automated. However, due to net movement caused by currents and frequent image breakups, its counter and sizer are not ideal for measuring farmed fish. Using DIDSON imaging, this research proposes a completely automated acoustic approach for calculating and evaluating farmed fish during fish transfer. After establishing the image with an image phase-only correlation approach, the background is subtracted. The fish is segmented using a contour tracing method that involves tracing the edges. A Kalman filter technique was devised and adapted to forecast fish migrations to avoid retelling the same fish. The track's spatiotemporal course was analyzed to perform automated counting. The body length of the fish was calculated by adding the centerline segments from the head to the tail of the fish. The proposed approach was tested on farmed yellowtail, Secondary source quinqueradiata mean total length 83.1 cm, with a mean total length sizing error of 2.4 cm. (Han, Honda, Asada and Shibata, 2009)

To resolve issues, this research provides a real-time, precise, unbiased, and profitable fish population counting approach based on a hybrid neural network model in far offshore fish mariculture. To collect the key information of different receptive fields, a multicolumn convolution neural network (MCNN) is used as the front end. Involved in resolving of various sizes are employed to adapt to angle, shape, and size variations induced by fish motion. Simultaneously, to limit the loss of spatial structure information during network transmission, a broader and deeper dilated coevolutionary neural network (DCNN) is used at the back end. (Zhang et al., 2020)

In this article, a smart tank has been designed with the problems of those who are unable to maintain their aquarium daily in mind. The feeder is driven by a servomotor that feeds the fish when a button is pressed, the heater regulates the temperature of the aquarium, and we can also control the fluorescent lamp, which helps in the growth of fish and plants. The proposed system employs a Raspberry Pi Webcam server to collect a live video stream of the aquarium, a servomotor to feed fish using pulse-width modulation, relays to control light, temperature, and filtering, and an Android app to give the user access to numerous functions. (Kori, Ayatti, Lalbeg and Angadi, 2018)

Chapter 03

3.1 Planning

This project is to design and build an automated aquarium that will allow caregivers to monitor their fish and aquarium on a daily basis while keeping the manual to a minimum. All aquarium-related functions, such as temperature control, automatic feeding, water change, water level monitoring, PH level monitoring and population monitoring will be monitored.

The solution we propose in this paper makes use of embedded devices, the Internet of Things, and deep learning to optimize fish farm management. Five modules are depicted in the system block diagram.

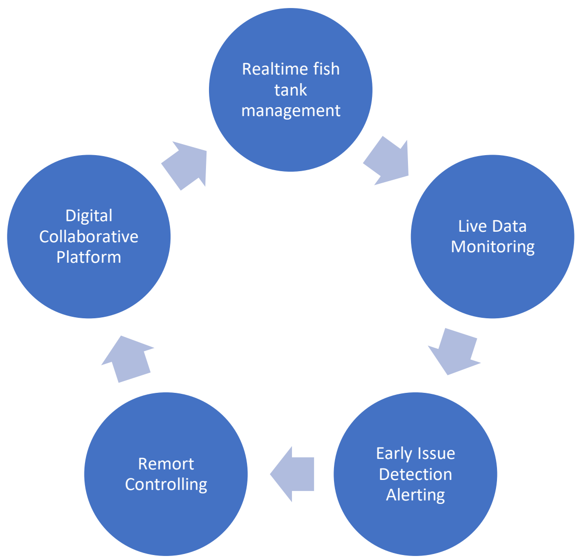


Figure 1 - System block diagram

Module 1 continuously analyses the water quality parameters using a range of sensors. Real-time sensor data is delivered to the farmer's mobile phone. Module 2 provides fish growers with real-time 2D graphical displays of several water quality metrics. Module 3 offers early and automatic problem detection and alerts the farmer via mobile device to any anomalies or low-quality detection in the fish tank. Module 4 enables remote operation of a variety of pumps, including oxygen and water heater/cooler pumps, as well as feeder pumps. Finally, Module 5 provides access to a digital community for capacity development, market development, and collaboration among fish producers.

3.1.1 Identifying Business Values

A business's total value is defined as its business value. As a result, a broad range is described when examining the commercial value of an IoT-based pet aquaculture management system. Commercial value of "pet fish" varies. This will enable farmers to balance the living standards of urban society in their daily activities and achieve the smart city goal.

- At present, there are a large number of part-time fish farmers in Sri Lanka. They have very little time to devote to fish farming along with their daily work. So, with this IOT pet fish farming management system, they can easily manage all the tanks.

- This method enables the identification of all current environmental conditions that are suitable for fish tanks. Additionally, it has the potential to significantly reduce fish mortality. As a result, such a system will significantly aid both small-scale and large-scale fish farmers in growing their businesses.

- Also, due to the lack of time, farmers who like to cultivate fish can easily manage their activities and such systems increase the number of people who are attracted to fish farming.

3.1.2 Feasibility Analysis

3.1.2.1 Technical Feasibility

This includes improving the IoT based Pet Fish Farming Management System in testing the potential for, increased efficiency and increased profitability. In addition to economic motivations, improving fish farming management was also a significant driving factor. In addition, the information on fish farming placement courts can be restructured. The purpose of this study was to introduce the concept of IoT to improve the Fish Farming Management System.

Hardware Requirement

- Liquid Water PH Value Detection Sensor with Control Board

Hydrogen power is a measure of how many hydrogen ions are in the body. The pH stands for hydrogen power. This is used to check the water quality and to grow fish and other seafood. There are 14 points on the pH scale, from 1 to 14, with 7 being neutral. A solution that is acidic has a pH of less than 7, and one that is very basic has a pH of more than 7. It's easy to connect the PH electrode to a pH meter, controller, or any other pH device with a BNC input terminal because there is only one cylinder on the electrode. The pH electrode probe is very accurate and reliable, so it can take measurements that aren't too far apart.

Figure 2 - Liquid Water PH Value Detection Sensor with Control Board

- ESP32 OV2640 Camera Bluetooth Wi-Fi Board

The ESP32-CAM is a smaller, low-power camera module based on the ESP32 microcontroller. It features an OV2640 camera and a TF card slot onboard. Wireless video surveillance, WiFi image upload, QR identification, and other intelligent IoT applications can all benefit from the ESP32-CAM.

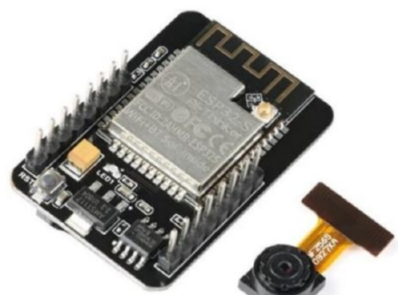


Figure 3 - ESP32 OV2640 Camera Bluetooth Wi-Fi Board

- Digital Waterproof Temperature Probe DS18B20 100CM

The single wire protocol digital temperature sensor, such as the DS18B20, can be used to measure temperature in the range of -67oF to +257oF (-55oC to +125oC) with +-5 percent accuracy. The data received from the 1-wire can be in the 9-bit to 12-bit range. Because this sensor adheres to the single wire protocol, it may be controlled by a single pin on a microcontroller. This is an advanced level protocol in which each sensor may be programmed with a 64-bit serial code, allowing multiple sensors to be controlled from a single microcontroller pin

Figure 4 - Digital Waterproof Temperature Probe DS18B20 100CM

- NodeMCU WIFI Internet Development Board ESP8266 CP2102

    The ESP8266 NodeMCU development board with cp2102 is a true plug-and-play option for low-cost WiFi projects. The module is pre-flashed with NodeMCU firmware, so all you must do now is install the USB driver. The NodeMCU project is open-source, and you can get all the design files there. The Arduino IDE programming makes it easy to program this microcontroller board.

Figure 5 - NodeMCU WIFI Internet Development Board ESP8266 CP2102

- Mini Micro Submersible Water Pumps DC 2.5-6V Amphibious Type

    DC 3V 4.5V 5V 6V Brushless Motor Horizontal Vertical Amphibious Type Micro Submersible Water Pump 120L/H Mini Pumps White Pump without power supply, power supply matched to the other, and water pump is a dc pump, pump voltage 3 v, use 6 v power supply with DC3

Figure 6 - Mini Micro Submersible Water Pumps DC 2.5-6V Amphibious Type

- Servo Motor SG90 9G Micro Servo Motor

If you want a small and light computer motor that can move things, this is the right one for you. The motor can move around 180 degrees and works in the same way as a normal servo, but it is smaller than a normal one. In order to move these servos around, you can use any servo code, hardware, or library that works with them. I think it's great for people who want to move things but don't want to build a motor controller with inputs and a gearbox, because it fits in small places.

Figure 7 - Servo Motor SG90 9G Micro Servo Motor

- Water Level Float Switch

A float switch is used to determine the fluid level in a tank or container. It floats above the liquid surface and acts as a mechanical switch in response to changes in the liquid level. They operate devices such as pumps (which bring in or take out water), valves (which open or close inserts or copies), and alarms, which notify users.

Figure 8 - Water Level Float Switch

- Raspberry Pi

In the United Kingdom, the Raspberry Pi is a line of small single-tablet computers made by Broadcom and made with help from people in the country. The original goal of the Raspberry Pi project was to help people learn basic computer science in schools and in countries that don't have enough money to buy computers. People all over the world use the Raspberry Pi to learn how to code, build hardware projects, automate their homes, set up Kubernetes clusters, use edge computing, and even do industrial work.

Figure 9 - Raspberry Pi

Software requirement

- Arduino Language

Figure 10 - Arduino

The Arduino IDE is a computer program that lets you write sketches (the Arduino equivalent of a program) for a wide range of Arduino boards, like the Arduino Due. Processing is a hardware programming language that is very similar to C. It is the foundation of the Arduino programming language.

- Blynk Server

Figure 11 - Blynk Server

The Blynk platform enables low-cost manufacturers of smart home products, advanced HVAC systems, agricultural equipment, and everything in between. These businesses create branded applications without writing any code and receive complete lagging IoT infrastructure with a single subscription.

- Fritzing

Figure 12 - Fritzing

Fritzing is an open-source initiative that aims to develop CAD software for amateurs or hobbyists interested in designing electronic hardware, assisting designers and artists in preparing to build a more stable circuit through prototype experimentation.

- Python

Figure 13 - Python

Python is a well-defined, general-purpose programming language with a high level of abstraction. Its design philosophy places a premium on readability through the use of significant indentations. Its language structure and object-oriented approach are intended to enable programmers to write concise, logical code for both small and large-scale projects.

- Open CV

Figure 14 - Open CV

OpenCV is a collection of programming functions designed primarily for real-time computer vision. Developed initially by Intel, it was later backed by Willow Garage and Itseez. The library is cross-platform and open source, with the Apache 2 license allowing for unrestricted use.

- SQLite

Figure 15 - SQLite

SQLite is a C library that contains a connection-based database management system. SQLite, in contrast to a large number of other database management systems, is not a client-server database engine. Rather than that, it is incorporated into the final program.

3.1.2.2 Financial Feasibility

This part allows you to analyse the costs and advantages of the project and determine whether it can be completed economically. Although this is also linked to producing income, there should be some benefit to owning and that justifies the expense of construction. And, because the advantages are never infinite, developers have a responsibility to create a system that meets the requirements at the lowest possible cost.

Products	Budget
Liquid Water PH Value Detection Sensor with Control Board	Rs.4300.00
ESP32 OV2640 Camera Bluetooth Wi-Fi Board	Rs.1790.00
Digital Waterproof Temperature Probe DS18B20 100CM	Rs.310.00
NodeMCU WIFI Internet Development Board ESP8266 CP2102	Rs.980.00
Mini Micro Submersible Water Pumps DC 2.5-6V Amphibious Type	Rs.280.00
Water Level Sensor Detection Liquid Surface Depth Height	Rs.120.00
Servo Motor SG90 9G Micro Servo Motor	Rs.145.00
Raspberry Pi	
Total Price	**Rs.7,934.00**

Table 1 - Financial Feasibility

3.1.2.3 Operational Feasibility

It is necessary to test the operational feasibility. There should be no issues with it being functioning. This indicates that the system is operational to the fullest extent possible. In addition, this strategic alignment analysis evaluates the project's appropriateness with the corporate plan while minimizing risk as the project grows in institutional feasibility. In addition, the system boosts product productivity and makes it easier to use. If wise and effective techniques are implemented to maintain this system safe, the advantages can be enjoyed without discomfort. In addition. The system's user capability should be evaluated.

3.1.2.4 Legal Feasibility

The process of determining whether a proposed project meets legal and ethical requirements is called legal feasibility. It is important to follow the requirements for starting a business or project, including business licenses, certificates, copyright, business insurance, tax identification number, health and safety measures, and more.

3.1.2.5 Schedule Feasibility

This project will last nearly eight months, beginning on July 11, 2022 and concluding on February 13, 2022. Approach, Preparation Implementation and Completion and submission are the four basic phases of this project. Thus, adhering to the Gantt chart can provide additional nuance and sub-exercises on acceptance.

No	TASK	START	END
	Step 1- Approach	**04/07/2021**	**25/07/2025**
1.	Define project title and supervisor allocation	04/07/2021	11/07/2021
2.	Research Papers Study	04/07/2021	24/07/2021
3.	Project proposal and Presentation	18/07/2021	25/07/2021
	Step 2 - Preparation	**26/07/2021**	**25/08/2021**
4.	Draw the System design	26/07/2021	30/07/2021
5.	Design the Project	30/07/2021	15/08/2021
6.	Choose and gather hardware component	16/08/2021	24/08/2021
	Step 3 - Implementation	**24/08/2021**	**30/12/2021**
7.	Develop the system	24/08/2021	28/11/2021
8.	Implement hardware structure	29/11/2021	25/12/2021
9.	Run and test for errors	25/12/2021	30/12/2021
	Step 4- Completion and submission	**31/12/2021**	**13/02/2022**
10.	Project report preparation	31/12/2021	25/01/2022
11.	Final prototype and design testing	26/01/2022	30/01/2022
12.	Conclusion and evaluation	31/01/2021	12/02/2022
13.	Delivery of the project	13/02/2022	13/02/2022

Table 2 - Schedule Feasibility

3.1.3 Work Plan

The project is divided into four phases:

- Approach
- Preparation
- Implementation
- Completion and Submission

It is primarily concerned with the assignment of a project title and supervisor, the study of a research paper, and the preparation of a project proposal and presentation. This requires a minimum of four weeks, and the project design phase begins with the submission and approval of the project proposal. This process takes approximately three weeks and is followed by the selection of hardware components. It lasts two weeks, and the COVID-19 condition may cause some time changes during this time. The project's implementation begins with the selection of hardware components. The entire project is expected to take approximately four months to complete. The report's preparation will begin when the project reaches its conclusion and will take approximately three weeks.

Task Name	Duration	Start Date	End Date
Step 1 - Approach		**7/4/2021**	**7/25/2021**
Define project title and supervisor allocation	1 week	7/4/2021	7/11/2021
Research Papers Study	5 days	7/4/2021	7/24/2021
Project proposal and Presentation	1 week	7/18/2021	7/25/2021
Step 2 - Preparation		**7/26/2021**	**8/25/2021**
Study the draw design	2 days	7/26/2021	7/30/2021
Design the Project	1 week and 5 days	7/30/2021	8/15/2021
Choose and gathet hardware component	2 weeks	8/16/2021	8/24/2021
Step 3 - Implementation		**8/24/2021**	**12/30/2021**
Develop the program	13 weeks and 3 days	8/24/2021	11/28/2021
Implement hardware structure	2 weeks and 2 days	11/29/2021	12/25/2021
Run and test of errors	4 days	12/25/2021	12/30/2021
Step 4- Completion and Submission		**12/31/2021**	**2/13/2022**
Project report preparation	4 weeks and 4 days	12/31/2021	1/25/2022
Final prototype and design testing	1 week	1/26/2022	1/30/2022
Conclusion and evaluation	1 week and 5 days	1/31/2022	2/12/2022
Delivery of the project	1 day	2/13/2022	2/13/2022

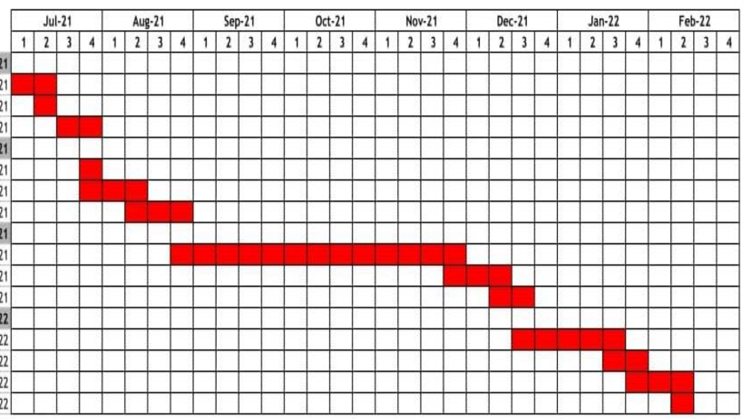


Table 3 - Gantt Chart

3.2 Analysis and Requirement Gathering

3.2.1 Analysis

The system has been developed under five phases to develop a more suitable system for IoT based Pet Fish Farming Management System. They are initialization, information gathering, analysis and planning, testing and development, and implementation. Also, these five components use different strategies to design a system that is easy to use and accessible. The aim of gathering requirements for the project of "IoT based Pet Fish Farming Management System" For my project, I'm going to use surveys as the main way to get information. Because surveys are more convenient and cost-effective than other ways to get people's answers, I use them as the main way to get data for my project.

3.2.2 Requirement Gathering

Gathering needs is a very important step in creating a project. We must meet the requirements for IoT based Pet Fish Farming Management System. Methods of collecting requirements that can be found in the industry. They are as follows,

- Interview
- Questionnaires
- Observation

Conducted through a questionnaire to gather information needed for current fish farming methods and new systems. Selected fish farmers were randomly selected and presented with a questionnaire. The questionnaire consisted of eleven questions, all of which needed to be answered. The questionnaire was created using Google Forms, and the link was sent to selected individuals via email and WhatsApp. The questions and responses are included below.

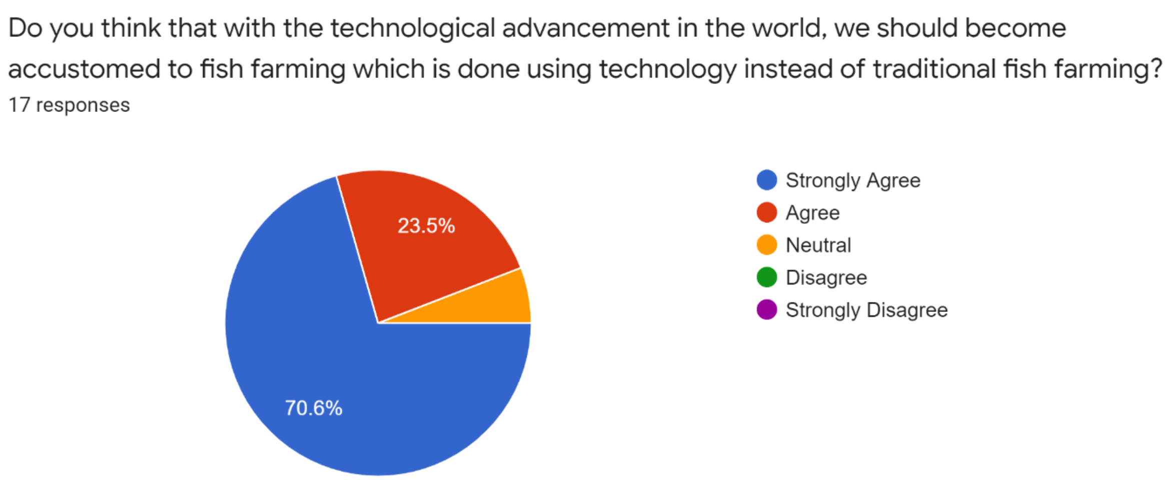


Figure 16 – Response – 1

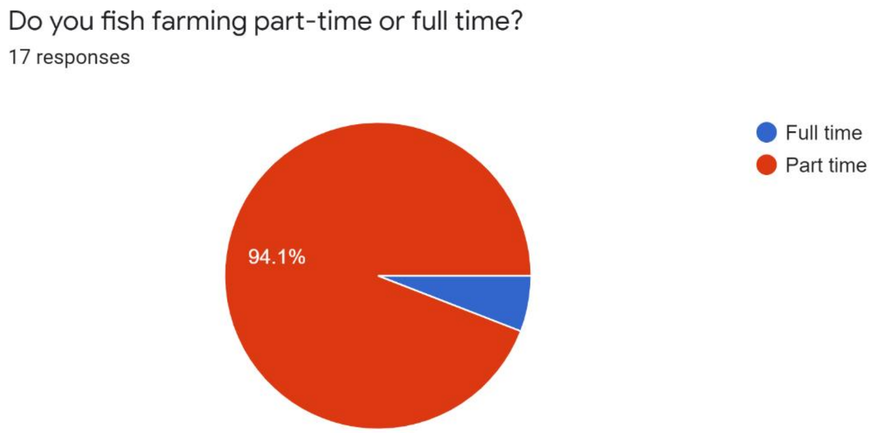


Figure 17 - Response – 2

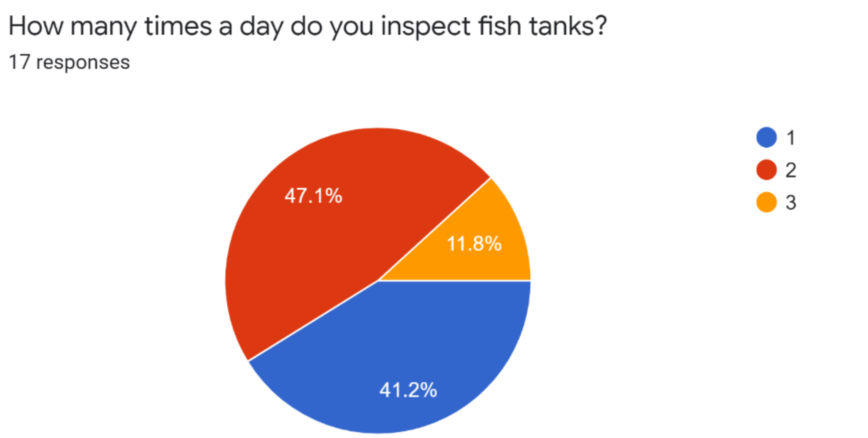


Figure 18 - Response – 3

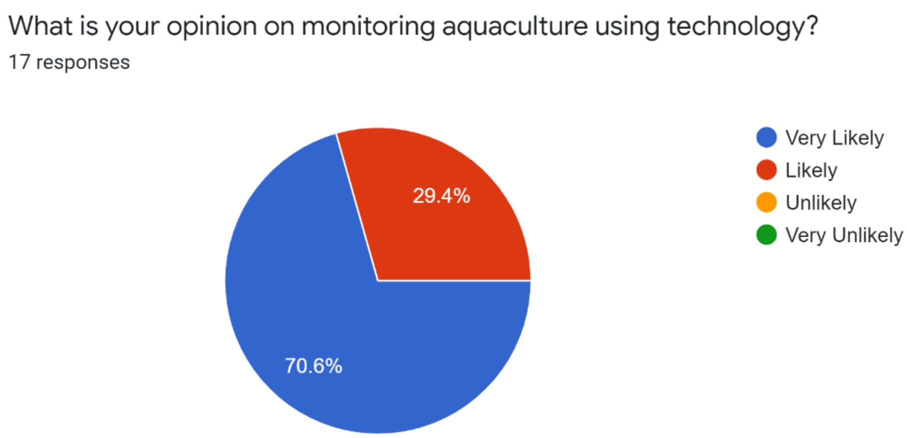


Figure 19 - Response – 4

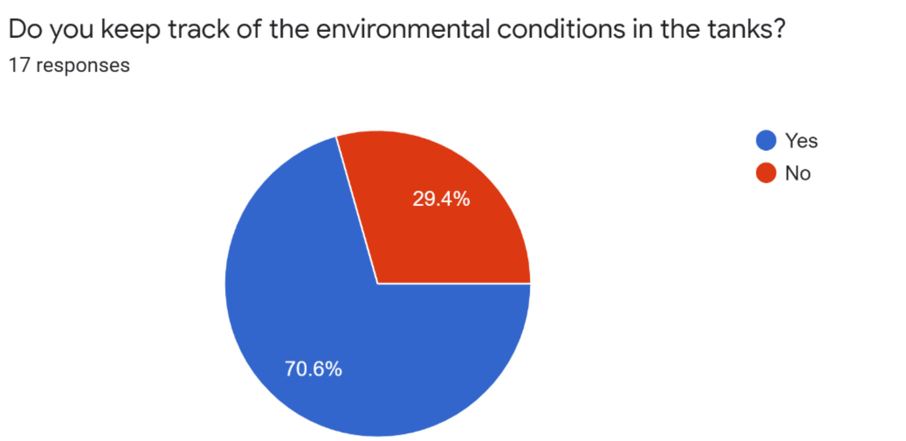


Figure 20 - Response – 5

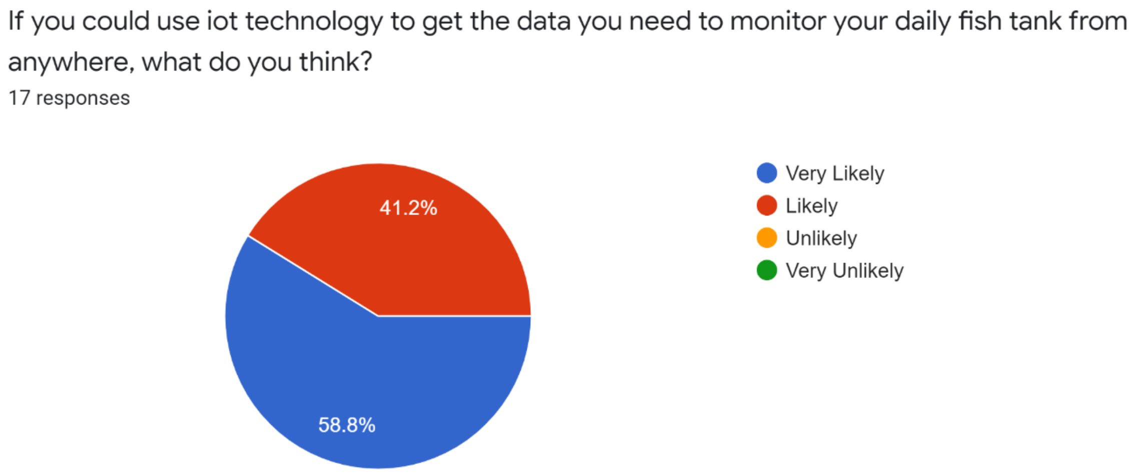


Figure 21 - Response – 6

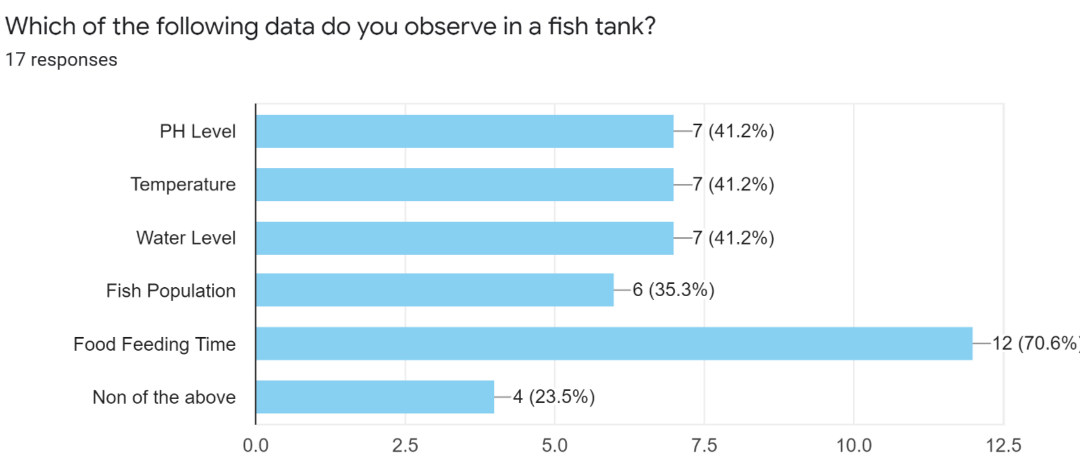


Figure 22 - Response – 7

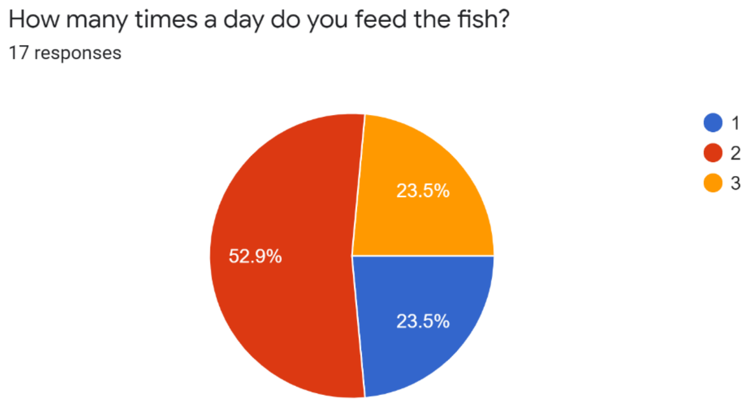


Figure 23 - Response – 8

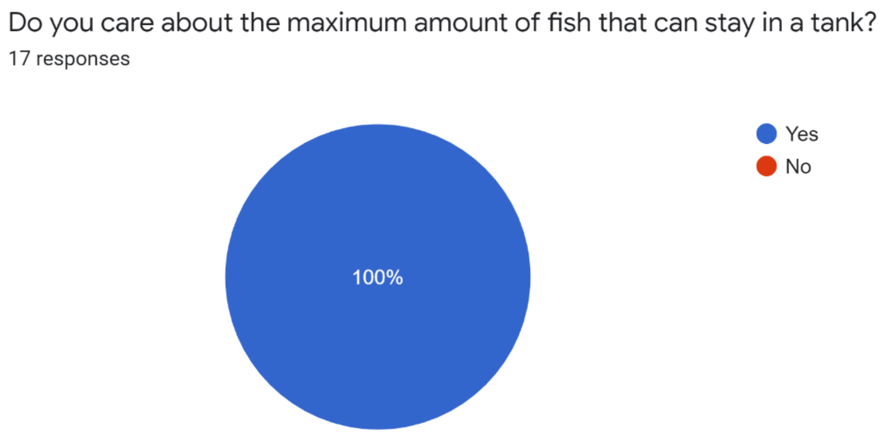


Figure 24 - Response – 9

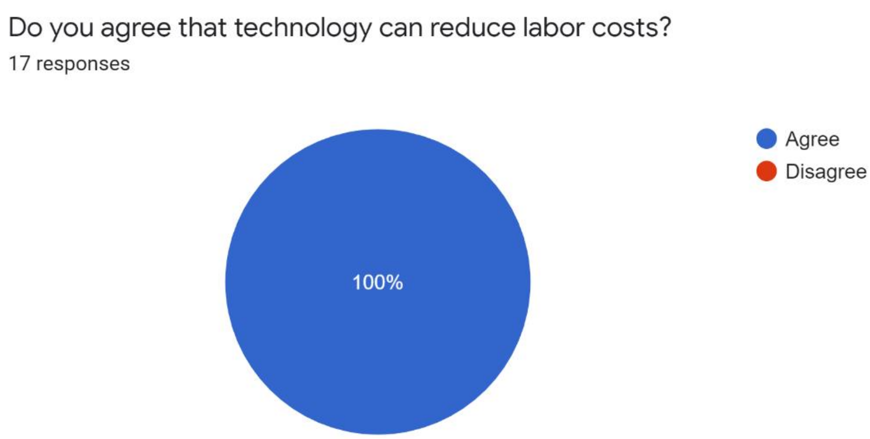


Figure 25 - Response – 10

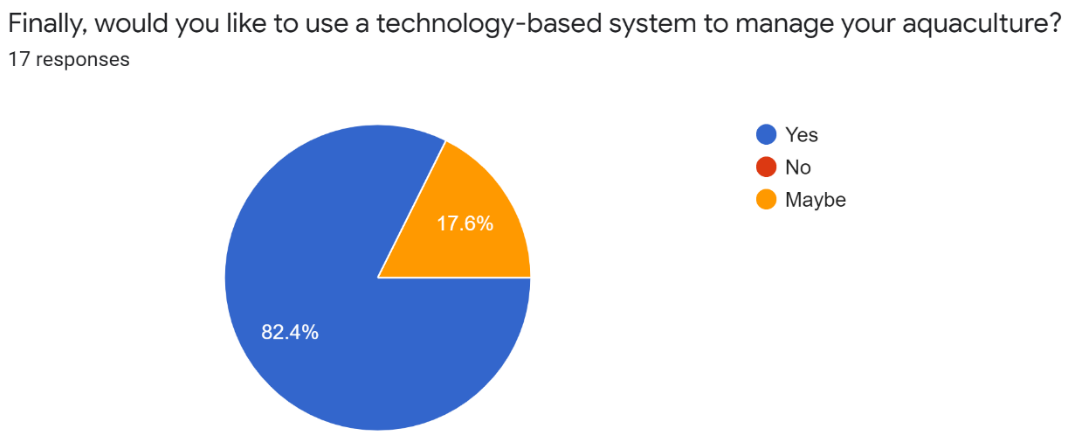


Figure 26 - Response – 11

The feedback I received from the quiz shows that more than 90% fish farmers do it part time. And about 70% of the work done part-time is done without even considering the environmental conditions of the fish tank. Therefore, it can be concluded that the number of fish dying farmers who cultivate fish part time is higher.

3.3 Designing

3.3.1 Physical Design

This system is very useful for fish farmers as it does the work of the farmers completely automatically. The main advantage here is that even though there are 5 or 6 tanks, the environmental conditions of all the tanks can be viewed from one place. Here, mainly fish farmers can access the mobile app to see the current temperature of the tank, the pH of the water at that time, and the amount of water currently in the tank.

There are two main types of this system. Below is the section that uses all the environmental data and other features of the fish tank from those two parts.

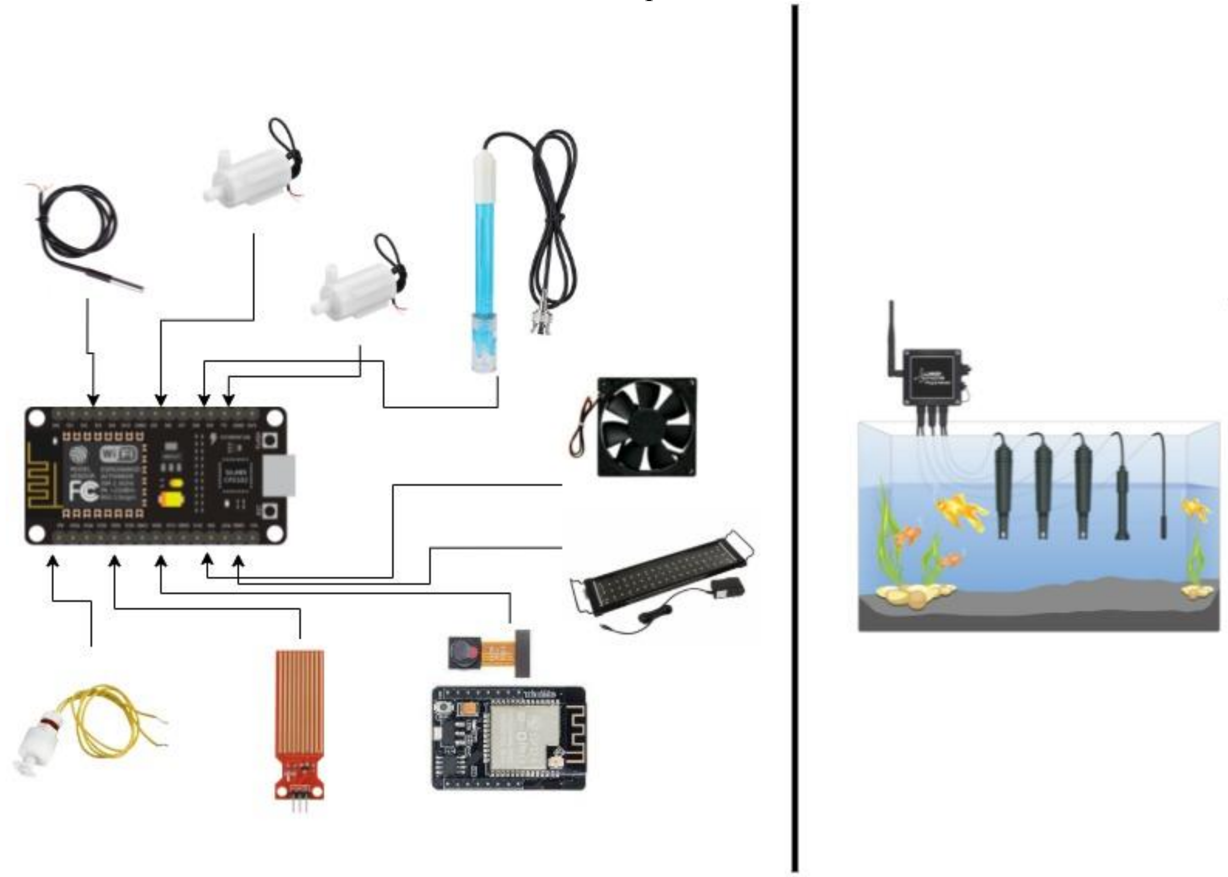


Figure 27 - Physical Design

The second part of this system is to calculate the fish population based on the images. Due to the diversity and complexity of the underwater environment, underwater images are usually subjected to intense noise, which reduces the quality of the underwater images and affects the accuracy of image analysis. Below is how the fish population is calculated based on the images.

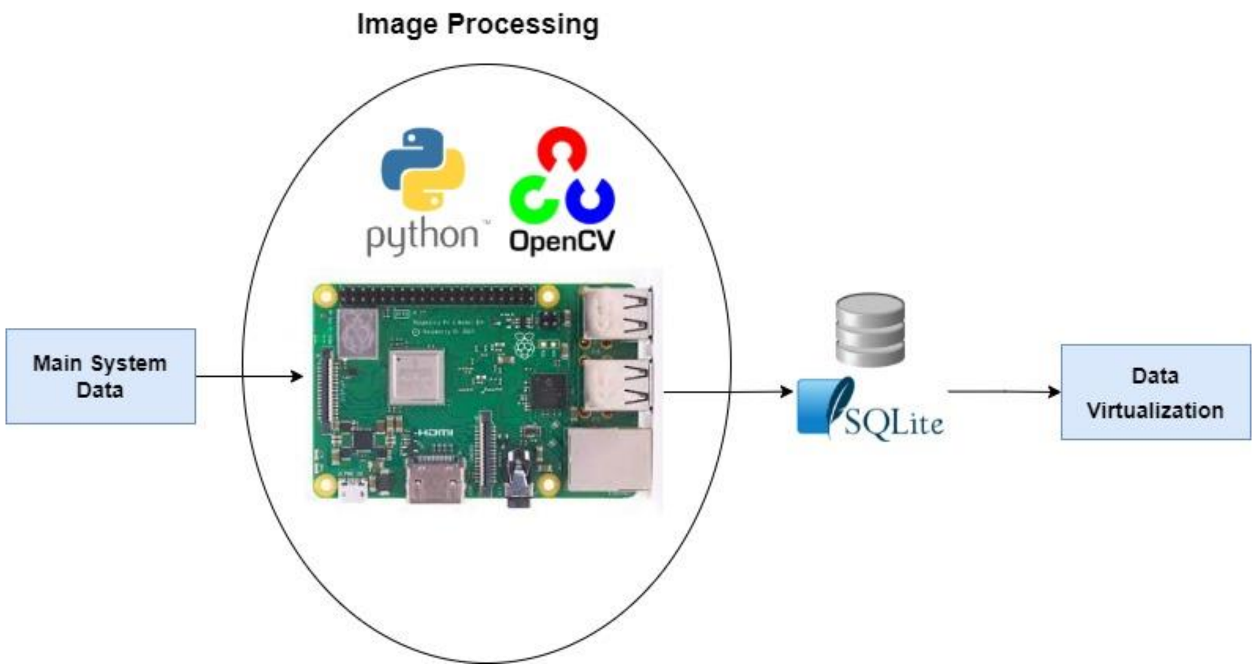


Figure 28 - Physical Design 1

3.3.2 Architecture Design

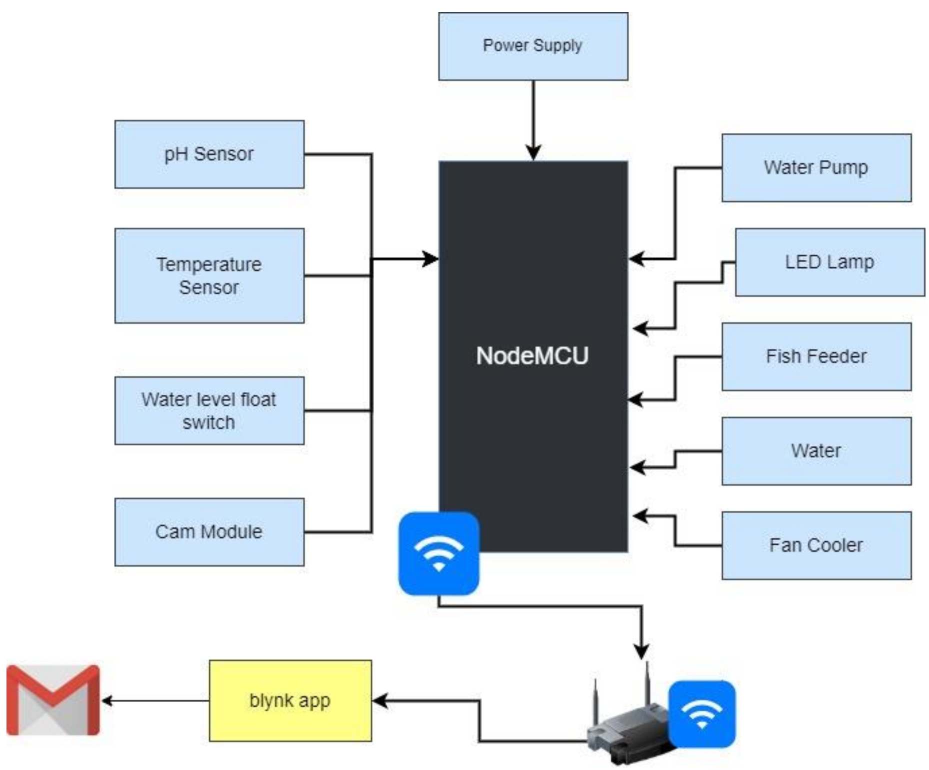


Figure 29 - Architecture Design

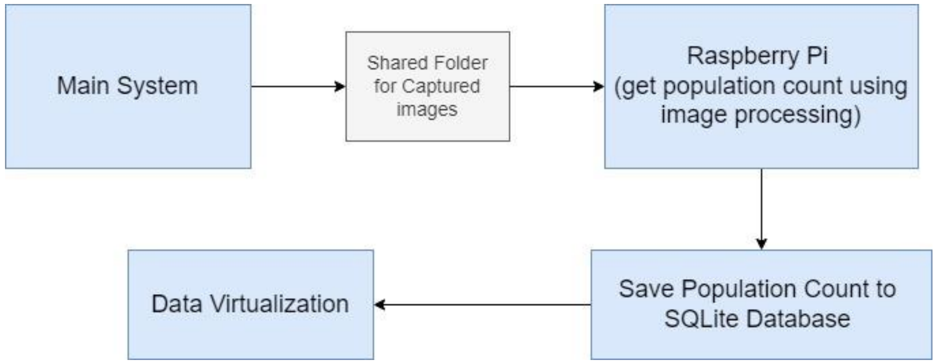


Figure 30 - Block diagram

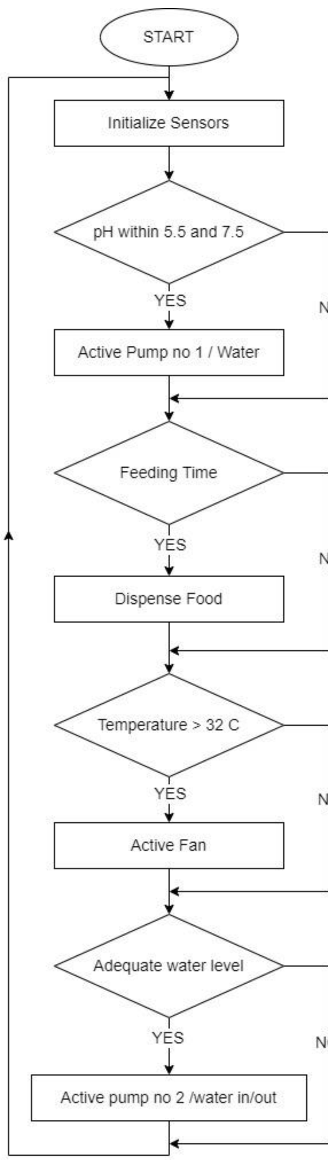


Figure 31 - Flow chart

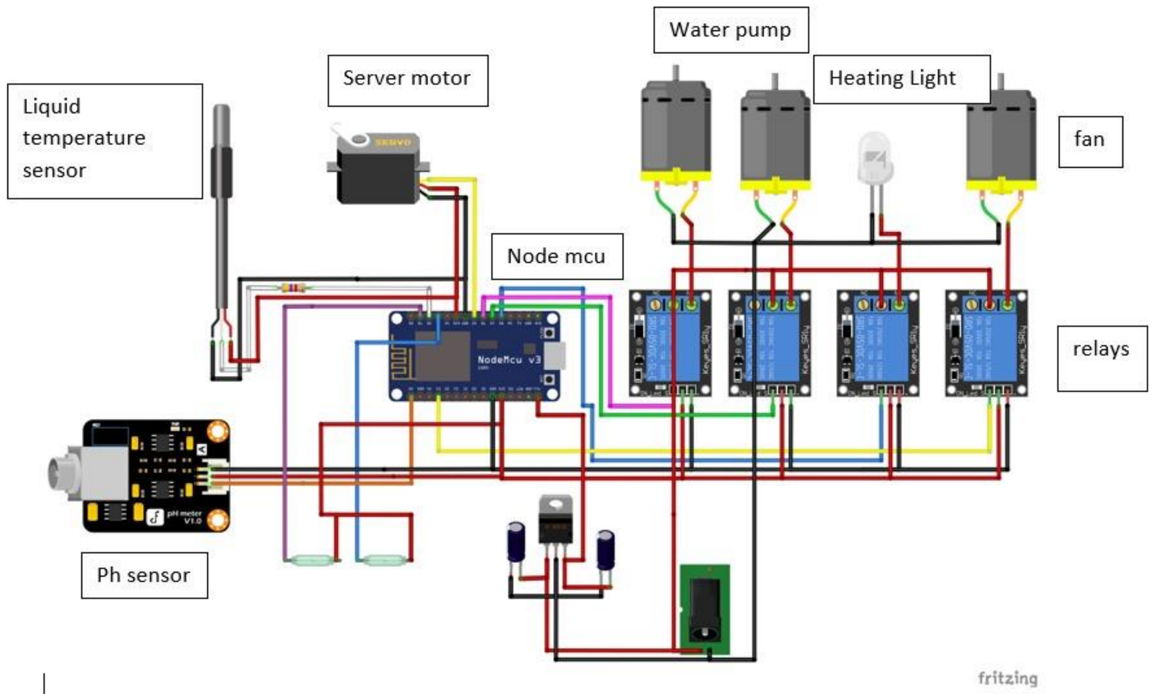


*Figure 32 – **Architecture Diagram***

3.3.3 Interface Design

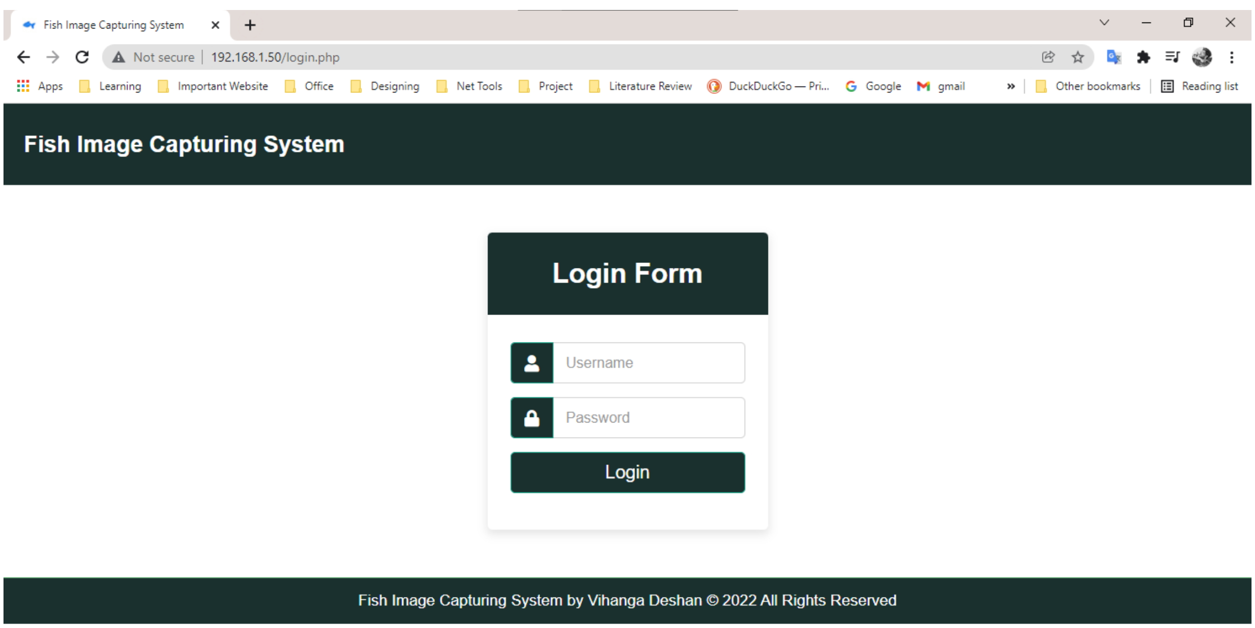


Figure 33 - Fish image capturing

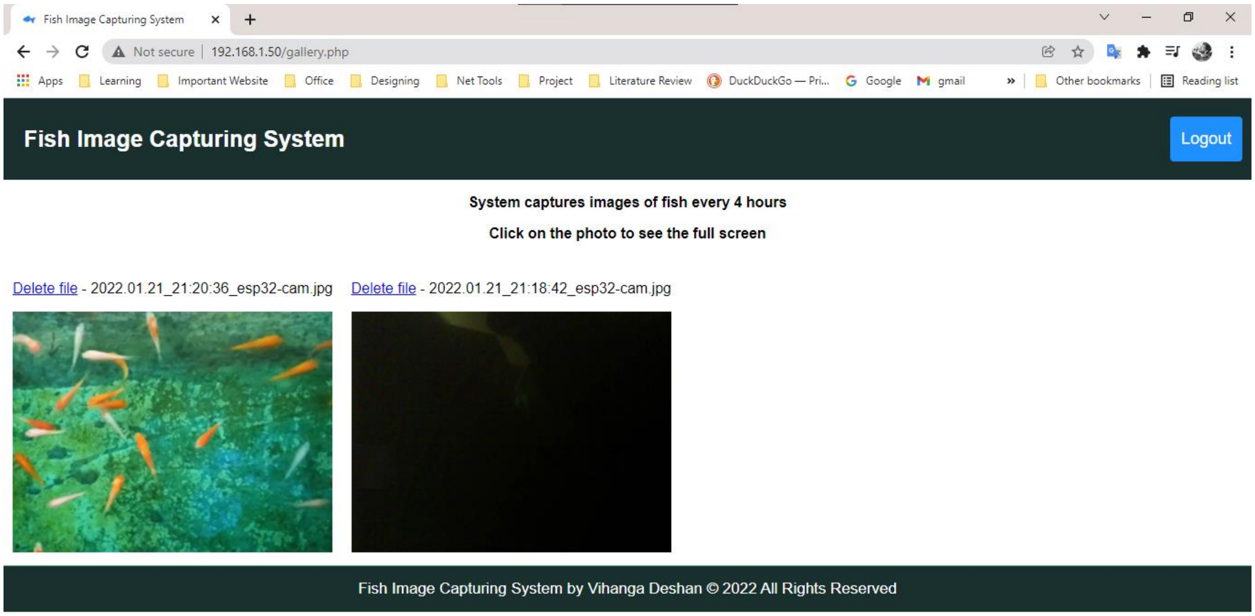


Figure 34 - Fish image capturing

3.4 Implementation

This system is mainly divided into two parts. The first part is the main part of the system where it controls all the environmental and other autoimmune conditions in the fish tank. The second part of the system is to calculate the number of fish in the tank on the basis of image processing and store the data in it and show the data virtualization from that data.

3.3 Testing

3.4.1 Unit Testing

Unit testing is a type of software testing that looks at how each piece of software works. The goal is to make sure that every piece of software code works correctly. Unit testing is done during the development code phase of an application. Unit tests are used to separate and check the correctness of a piece of code. A single function, method, process, module, or item can be called a unit, but you can also think of it as a whole.

- **PH Sensor Testing**

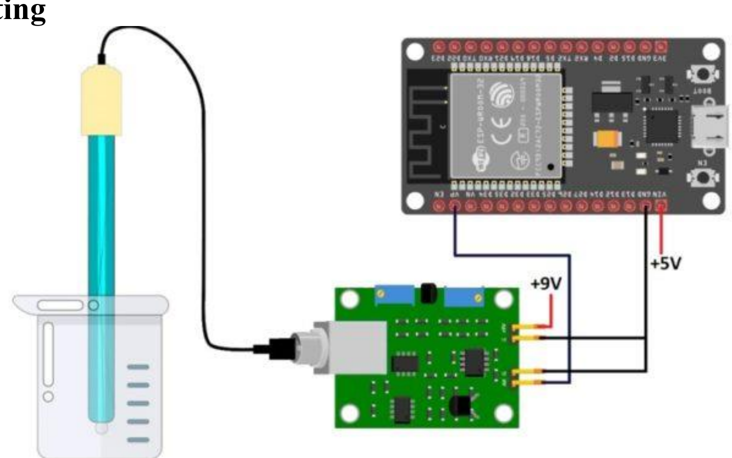


Figure 35 - PH Sensor Testing

Use an external 9V battery or a 9V DC supply to power the pH sensor. Connect the pH Sensor signal board's output pin to the VP pin on the ESP32, which can be used as an A0 pin. The sensor's output ranges between 0.5V and 3V, allowing it to be used with the ESP32's analog pins.

Basic Code -

```
const int potPin=A0;
float ph;
float Value=0;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(potPin,INPUT);
  delay(1000);
}
 void loop(){
    Value= analogRead(potPin);
    Serial.print(Value);
    Serial.print(" | ");
    float voltage=Value*(3.3/4095.0);
    ph=(3.3*voltage);
    Serial.println(ph);
    delay(500);
 }
```

Result of PH sensor testing –

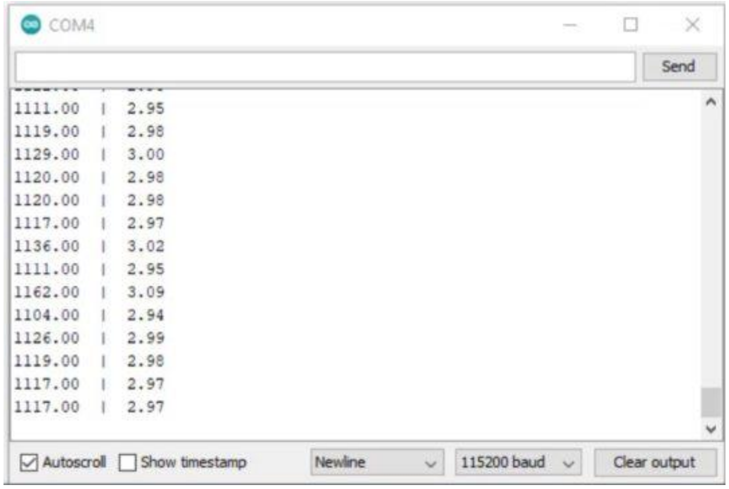


Figure 36 - Sensor result

- **Servo Motor Testing**

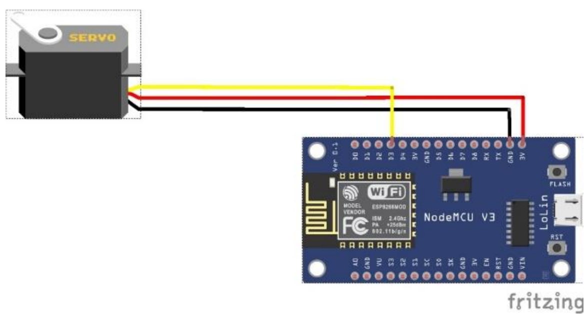


Figure 37 - Servo Motor Testing

Connect the red wire to the nodemcu's 3.3 volt pin. Connect the brown wire from the servo motor to the nodemcu's GND pin. Connect the servo motor's signal wire (orange wire) to the nodemcu's digital-3 pin. Once the connections are complete, power the nodemcu via a USB cable to start the servo motor.

Basic Code -

```
#include <Servo.h> // servo library
Servo s1;
void setup()
{
 s1.attach(0);  // servo attach D3 pin of arduino
}
void loop()
{
  s1.write(0);
  delay(1000);
  s1.write(90);
  delay(1000);
  s1.write(180);
  delay(1000);
}
```

- **DS18B20 Temperature Sensor Testing**

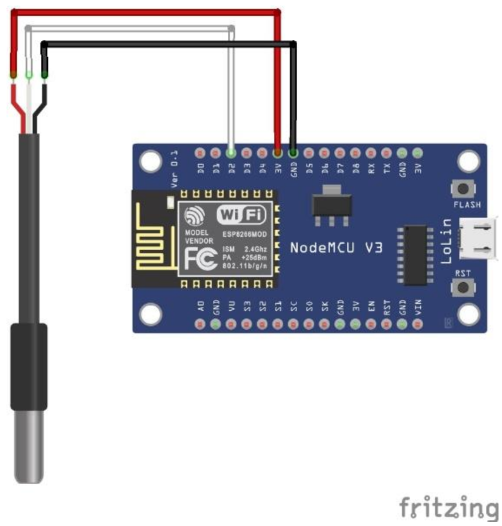


Figure 38 - DS18B20 Temperature Sensor Testing

The red wire should be connected to the 3v pin on the NodeMCU board, the white wire to the D2 pin, and the black wire to the GND pin. Also need to add the OneWire Library file to the Arduino Libraries for this DS18B20 Temperature Sensor to work.

Basic Code –

```
#include <DallasTemperature.h>
#include <OneWire.h>

#define ONE_WIRE_BUS 4                          //D2 pin of nodemcu

OneWire oneWire(ONE_WIRE_BUS);

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

void setup(void)
{
  Serial.begin(9600);
  sensors.begin();
}

void loop(void)
{
  sensors.requestTemperatures();                // Send the command to get temperatures
  Serial.println("Temperature is: ");
  Serial.println(sensors.getTempCByIndex(0));   // Why "byIndex"? You can have more than one IC on the same bus. 0 refers to the first IC on the wire
  delay(500);
}
```

Result of temperature sensor testing –

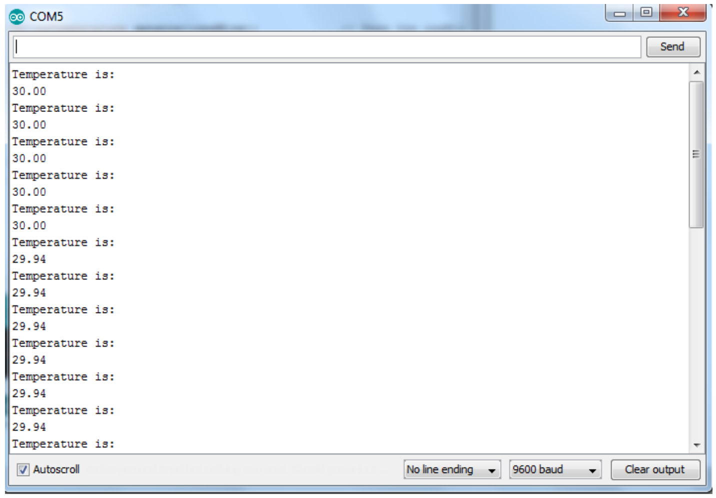


Figure 39 - DS18B20 Temperature Sensor Testing Result

- **Mini Water Pump Testing**

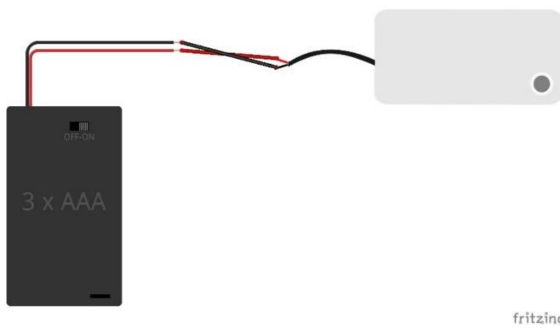


Figure 40 - Mini Water Pump Testing

A mini water pump requires a voltage of between 3 and 6 volts to operate. The red terminal of the water pump is connected to the battery's red terminal, while the black terminal is connected to the battery's black terminal.

Result of mini water pump testing
 -

Figure 41 - Mini Water Pump Testing result

- **ESP32 Cam Testing**

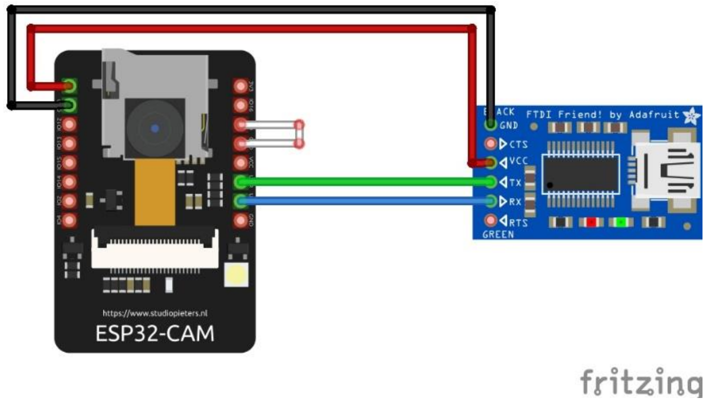


Figure 42 -ESP32 Cam testing

An FTDI programmer module is required to program the ESP32-CAM board. Here the GND pin of the ESP32-CAM board is connected to the GND of the FTDI module, the 5V pin to the VCC (5V) pin, the U0R pin to the TX pin, the U0T pin to the RX pin, and the GPIO 0 pin to the GND pin on the FTDI module. Should do. In this case the GPIO 0 pin should be connected to the GND pin of the FTDI module only when uploading a code. It must then be removed. The following code can be used to test this ESP32 Cam board.

Basic Code -

```
#include "esp_camera.h"
#include <WiFi.h>

// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#include "camera_pins.h"

const char* ssid = "SLT_FIBRE";
const char* password = "abc12345";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
```

```
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}
void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

Result of ESP32 Cam board testing –

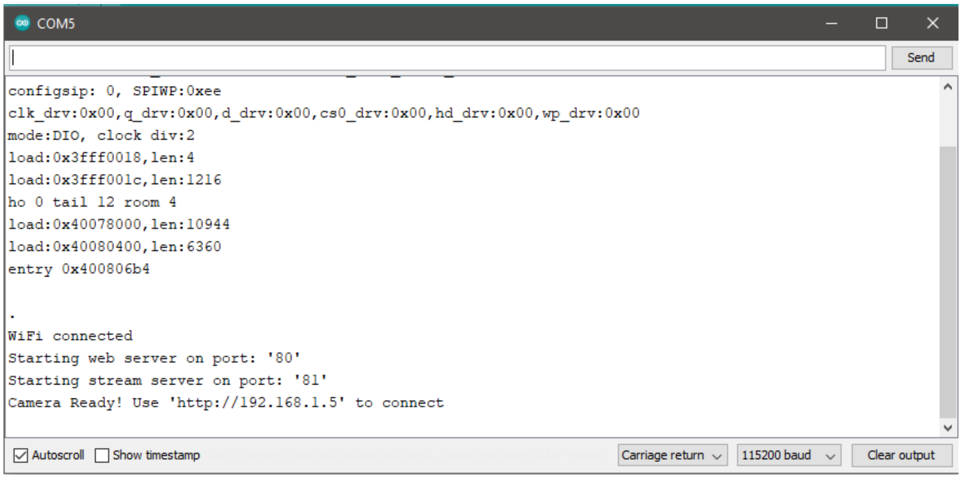


Figure 43 - Result of ESP32 Cam

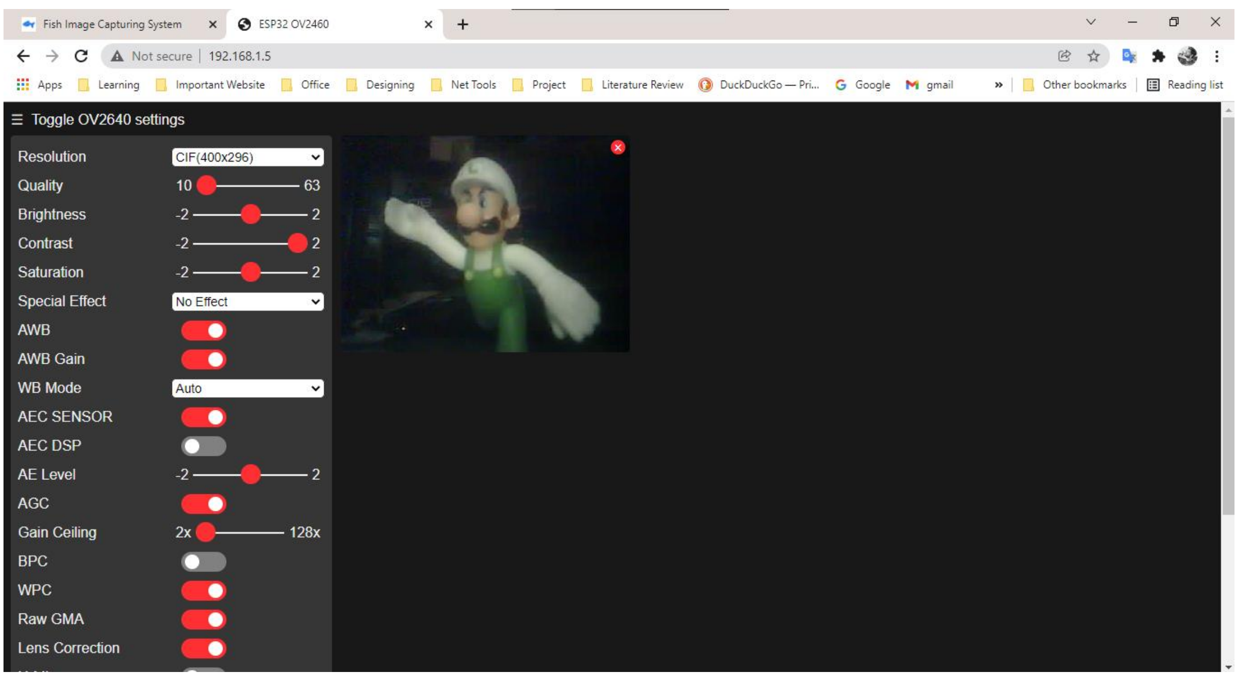


Figure 44 - Result of ESP32 Cam -1

- **Raspberry Pi Webserver Testing**

For fish farmers to use the Fish Image Capturing System and population counting features, a web server is required to store the required images and allow them to be viewed from anywhere on the web. I accomplish this through the use of a Raspberry Pi single-board computer. It also requires a web development language called php. The environment required to fulfill the two requirements is established, and the resulting data is presented below.

Apache2 web server testing -

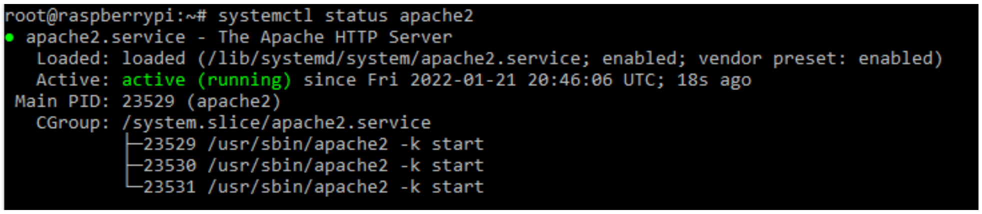


Figure 45 - Apache2 web server testing

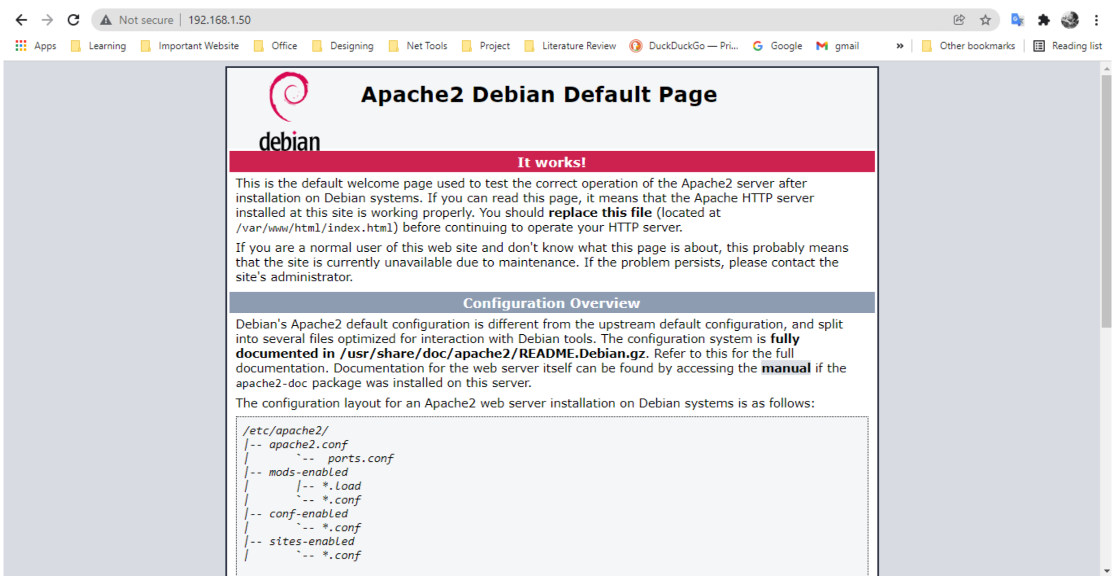


Figure 46 - Apache2 web server testing 1

Testing PHP language support –

```
<?php echo "hello world"; ?>
```

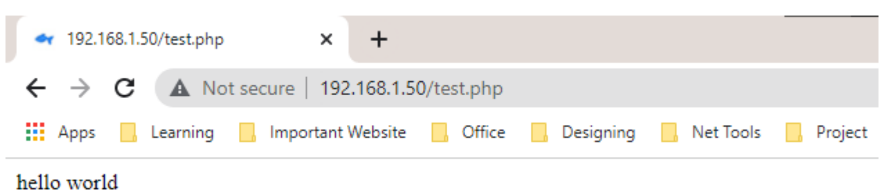


Figure 47 - Testing PHP language support

3.5.2 Integration Testing

Testing software together as a single unit is called integration testing, and it is a type of testing. A typical software project has a lot of different software modules made by a lot of different programmers. This stage of testing is meant to find problems with how different parts of software work together when they're combined.

- **PH Sensor and Temperature Sensor Testing**

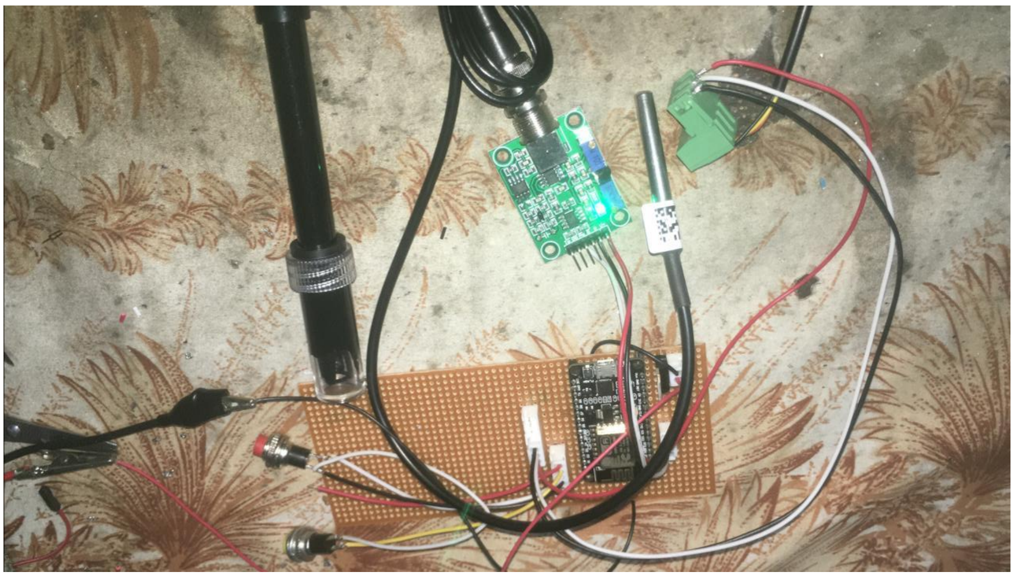


Figure 48 - PH Sensor and Temperature Sensor Testing

Result of PH and Temperature sensor testing –

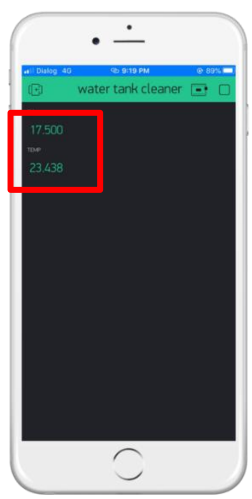


Figure 49 - Result of PH and Temperature sensor

- **ESP32 Cam Module and Raspberry Pi Web Server Connection Check**

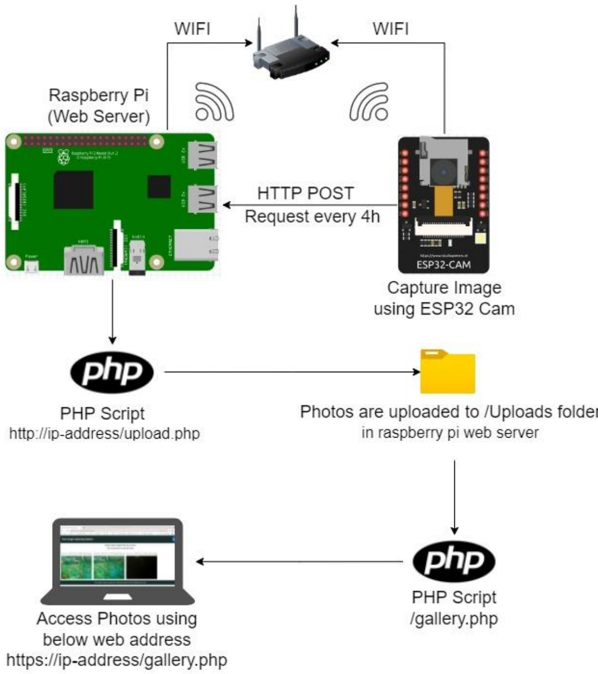


Figure 50 - ESP32 Cam Module and Raspberry Pi Web Server Connection Check

This is accomplished by sending the photos taken by the ESP32 Cam via an HTTP POST request to the Raspberry Pi's /uploads directory. That process is carried out via the upload.php PHP script running on the web server's port 80. After uploading the image to a directory called /upload, it can be viewed in a web browser via the gallery.php script on the web server at the URL called https://ipaddress/gallery.php. (All the relevant codes for this are given in the 4.2 sample code section)

Result of ESP32 Cam board and Raspberry-Pi web server connectivity check –

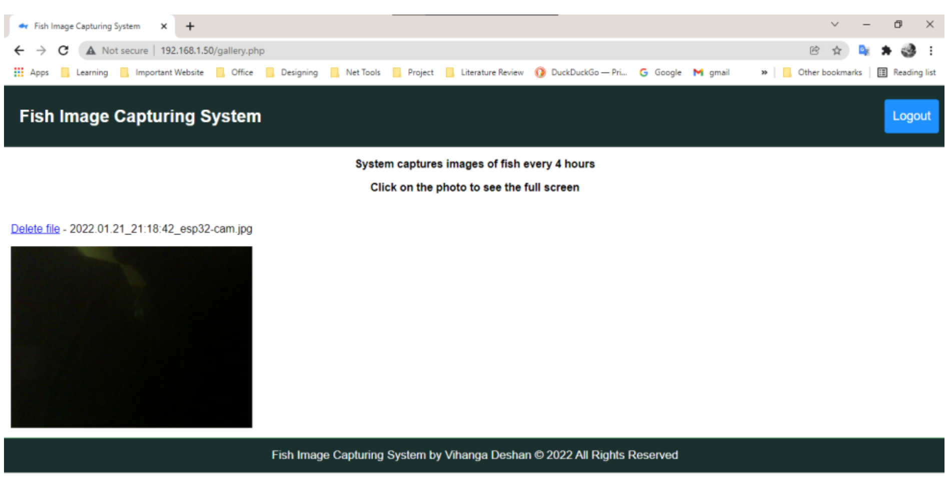


Figure 51 - Result of ESP32 Cam board and Raspberry-Pi web server connectivity

3.5.3 System Testing

System testing is a type of testing that looks at how well a piece of software works together. The last functional requirement is tested with a system test. Software is usually just one part of a bigger computer system. Eventually, the program will be able to work with a lot of different types of software and hardware. When a computer-based platform is put through its paces, system testing is the best way to make sure it works well. In the 4.1 Test Cases section, can find all of the parts of the system testing.

Chapter 04

4.1 Academic Findings

I believe the project's ultimate goal is to develop a system that allows medium-scale fish farmers to avoid the hassle of daily monitoring and to check the quality of their tanks from any location. Additionally, they are able to regulate the critical factors that the fish require in the tank. The following are the primary points to consider.

- How to properly inspect fishing tanks and see if there are any changes in the water that are harmful to all the fish?
- How to measure the PH level of the water is good for fish?
- How to remove drainage and refill the tank with fresh water?
- How can feed to the fish according to the pre-planned time?
- How to inform the farmer via SMS if the environmental conditions in the tank change adversely?
- How to calculate the fish population in the tank and maintain the tank under proper conditions?

The pH of the tank is a major factor in the survival of the fish. Numerous factors contribute to the increase in the pH of the fish tank. pH levels acceptable in the aquarium range from 6.8 to 7.6, with 7.0 being the median. This is a significant factor in the rise in pH of small-scale fish farmers. Inefficient or old filters can cause a variety of problems, but the most serious is that they do not match the pH of the water to the fish. Other organic matter that decomposes, such as fish waste and leftover food, increases the pH level. And the key is to maintain a low pH level. Once the old filter is replaced, more carbon monoxide becomes available, lowering the pH level. Because the majority of water filters are ineffective at removing fish waste, regular tank cleaning will ensure that your aquarium is completely free of contaminants. The pH monitoring feature in this system allows the farmer to monitor the pH value of the tank simultaneously via mobile phone.

The majority of fish grow best between 75- and 80-degrees Fahrenheit (23 and 27 degrees Celsius), but specific fish have different requirements. Natural temperature variations between day and night have a significant effect on the growth of fish with weakened immune systems, making them more susceptible to disease. As a solution to this too, the farmer can use his mobile phone to monitor the current temperature of the tank. The farmer can also control the temperature of the tank with the cooling fan and heater here

Another significant problem for small-scale fishmongers is the inability to feed the fish on time due to their other daily activities. Fish, naturally, will seek food when they are hungry, but will always consume food that they have. Consume food sources several times a day if they are plentiful. On the other hand, if food sources become scarce, fish may go days without being consumed. There is a separate part of this system for feeding fish and it can be controlled from the farmer's mobile phone.

The population of the fish in the tank also directly affects the growth of the fish. 1 gallon of fish requires approximately 2 gallons of water. This should be considered only to establish a fundamental understanding and sensitivity to the boundary. Excess fish can eventually result in disease due to stress. Here we can calculate the population from the photographs taken from the fish tank and if there are more fish in that tank than the required quantity, we can put it in another tank. This process is carried out independently of the grant system.

4.2 Sample Code

Fish Population Counting using Open CV

This code is used by the camera to track the number of fish in the image and print the number of fish on the terminal. This code is used to process the latest image in the directory. This imaging process is done mainly in 7 steps.

1. Convert image into grayscale
2. We must first blur the image to eliminate noise before attempting to recognize the edges.
3. Once the image is blurred, its edges can be identified using the canny algorithm.
4. Optimized the edges.
5. The number of fish can be obtained by converting the image from BGR to RGB and then drawing contours. But this algorithm is not 100% accurate.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

folder_path = r'C:\xampp\htdocs\j\uploads'
file_type = '\*jpg'
files = glob.glob(folder_path + file_type)
max_file = max(files, key=os.path.getctime)

image = cv2.imread(max_file)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("coins in the image : ", len(cnt))

```

Figure 52 - Sample coding 1

Upload Population Count into SQLite Database

Code 1.4 in the figure above indicates that the fish population can only be obtained using image processing. The code 1.4 in the figure below is used to store the number of fish in an SQLite database. Here only the timestamp and the current population of fish are uploaded to the database.

```
import cv2
import sqlite3
import datetime
import numpy as np
import matplotlib.pyplot as plt

conn = sqlite3.connect('fishpopulation.db')
c = conn.cursor()

def create_table():
    c.execute('CREATE TABLE IF NOT EXISTS FishPopulation (Date TIMESTAMP, Number REAL)')

currentDateTime = datetime.datetime.now()

folder_path = r'C:\xampp\htdocs\j\uploads'
file_type = '\*jpg'
files = glob.glob(folder_path + file_type)
max_file = max(files, key=os.path.getctime)

image = cv2.imread(max_file)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)

print("coins in the image : ", len(cnt))

def data_entry():
      date = currentDateTime
      number = len(cnt)
    c.execute("INSERT INTO FishPopulation (Date, Number) VALUES(?, ?)", (date, number))
    conn.commit()

    print("Data Insert successfully")
    create_table()
    data_entry()

c.close()
conn.close()
```

Figure 53 - Sample coding 2

Export SQLite Database into .csv file

The following code can be used to export the number of fish uploaded to the SQLite database as a .csv file. Here the exported .csv file is saved as data.csv in the same directory.

```
print("Exporting data into CSV............")

cursor = conn.cursor()
cursor.execute("select * from FishPopulation")
with open("data.csv", "w") as csv_file:
    csv_writer = csv.writer(csv_file, delimiter="\t")
    csv_writer.writerow(['Column 1', 'Column 2'])
    csv_writer.writerows(cursor)

dirpath = os.getcwd() + "/data.csv"
print("Data exported Successfully into {}".format(dirpath))

c.close()
conn.close()
```

Figure 54 - Sample coding 3

4.3 Test Cases

The following test cases test whether all parts of the system are working properly.

Test Case Identifier	**01**
Test Case	Checking the data obtained in the tank by the main system
Test Designed by	Vihanga Deshan
Test Designed Date	**2022.12.20**
Test Executed by	Vihanga Deshan
Test Executed Date	**2022.01.08**
Description	With this test case checks all sensors
Dependencies	Sensors, Modules

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
01	1.1	Check PH sensor			

Actual Result –
PH sensor reading view in graph

Screen Shot

Figure 55PH Test Case

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
01	1.2	Check Temperature Sensor	Fish tank temperature	Temperature display on the blynk app	Pass

Actual Result –
Display temperature value

Screen Shot

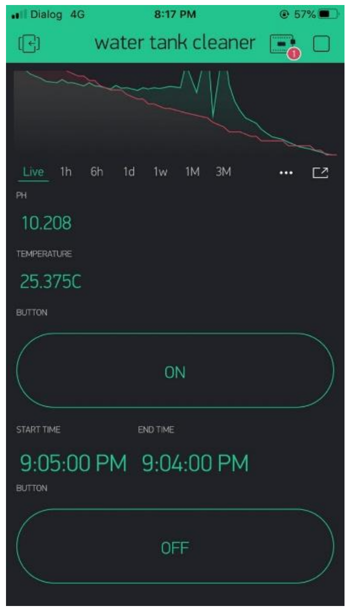


Figure 56Temperature Test Case

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
01	1.3	Check Report Generation via Blynk app	Temperature, PH	Temperature & PH Values should receive to email with csv format	Pass

Actual Result –
Received generated excel file

Screen Shot

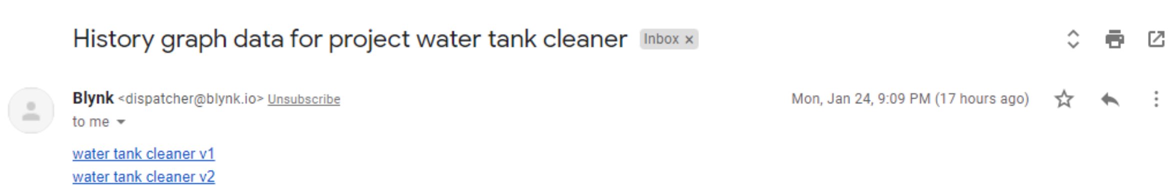


Figure 57History Data Reporting Test Case

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
01	1.7	Check Food Feeding System			

Actual Result –
Food Feeder turn on preschedule time plan

Screen Shot

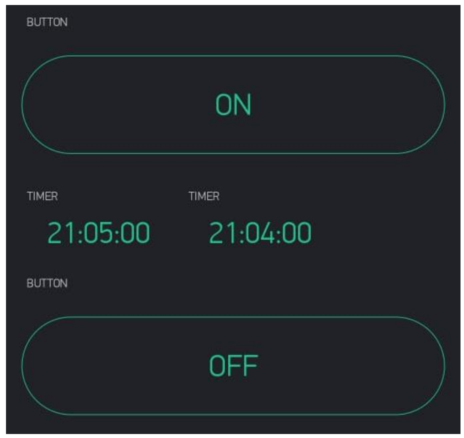


Figure 58 Food Feeding Timer Testing

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
01	1.10	Check image Capturing system	Image	ESP32 Camera Captured photos can view through the website	Pass

Actual Result –
Fish photos captured & uploaded to webserver

Screen Shot

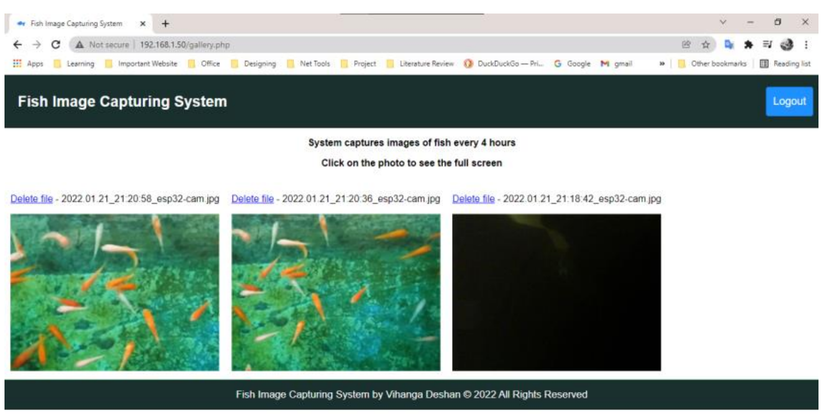


Figure 59 Image Capturing System Testing

Test Case Identifier	**02**
Test Case	Calculation of the number of fish present in the system using an image obtained from the cam module
Test Designed by	Vihanga Deshan
Test Designed Date	**2022.01.02**
Test Executed by	Vihanga Deshan
Test Executed Date	**2022.01.16**
Description	With this test case we can get the size of the fish in the tank according to that image.
Dependencies	Capture Image

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
02	2.1	See if the fish population can get through the code	Image	Fish Population Count	Pass

Actual Result –
Fish count in the image: 33

Screen Shot

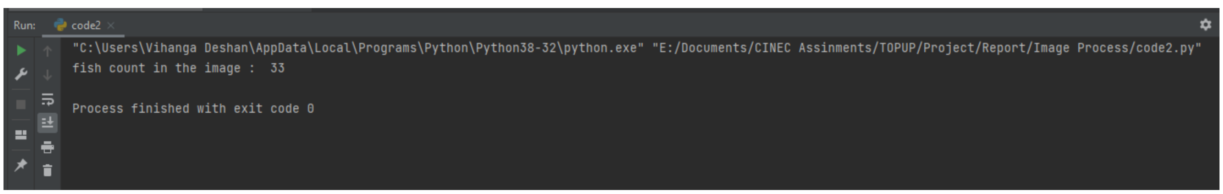


Figure 60 Fish population algorithm testing

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
02	2.2	Pass the Fish count data into SQLite Database	Image	Fish counts are insert into the database	Pass

Actual Result –
Fish count in the image: 33
Data Inset successfully

Screen Shot

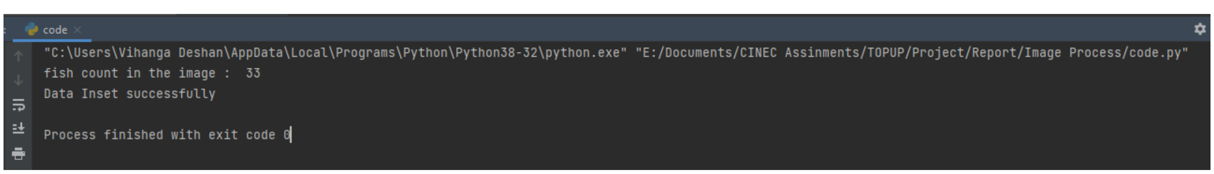


Figure 61 Connect SQLite Database

	Date	Number
	Filter	Filter
1	2022-01-16 13:34:41.715947	33.0
2	2022-01-16 16:12:40.377322	33.0

Figure 62 Database View

Test Case #	Sequence #	Test Case Description	Test Data / Input Value(s)	Expected Result	Status (Pass/Fail)
02	2.3	See if the population calculation code can execute through the website	Fish Image	View the fish count through and other print values can see in the web	Pass

Actual Result –

fish count in the image: 12
Data Insert successfully
Exporting data into CSV............
Data exported Successfully into C:\xampp\htdocs\j\python/data.csv

Screen Shot

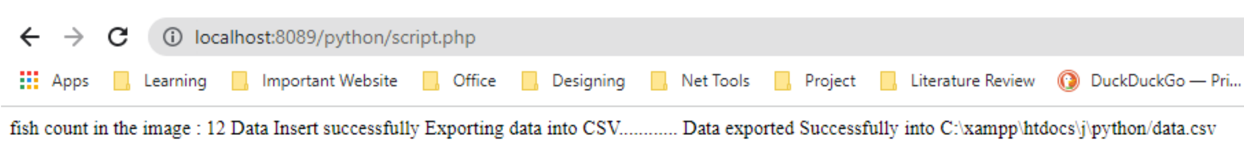


Figure 63 Python Script run using PHP

Chapter 05

5.1 Important Outcome

In conclusion, this project developed an IoT-based aquarium monitoring system. The primary objective of this system is to assist the aquarium owner in more conveniently and easily maintaining and monitoring the aquarium. This is a aquarium monitoring system that is dependable and capable of updating users on the current situation in real time.

Today, pet fish farming occupies a unique position in the world. The Internet of Things is being used to augment the capabilities of aquaculture automation systems. A Smart Fish Farming System built with cutting-edge technologies like Arduino and Wireless Sensor Networks. Aquaculture utilizes the Internet to increase production. And the majority of this business is conducted on a small scale. And they do so on a part-time basis. As a result, the amount of time they can devote to this business is extremely limited. This system enables them to remotely monitor and control the environmental conditions in their fish tanks. This is critical for small-scale farmers as well as large-scale farmers. Because as the number of tanks increases, the number of people required to maintain them increases proportionately. A system like this one developed through the Internet of Things can manage human labour and time.

Furthermore, the system can monitor the pH of the fish tanks, their temperature, their water level, automatic fish feeding, the removal of contaminated water from the tank, the addition of clean water, and live video recording of the tank from anywhere using a mobile phone. A unique feature of this tank is that the fish population can be captured by processing photo images captured during live video. However, the farmer must perform this task manually.

5.2 Limitation

Although the population of the fish in the tank can be obtained here, this has to be processed manually by the farmer. And here the fish population is not a prediction, so the farmer has to decide it according to their opinion. Additionally, caution must be exercised when processing high-quality images captured by the NodeMCU cam module in order to obtain an accurate population count.

5.3 Critical Evaluation

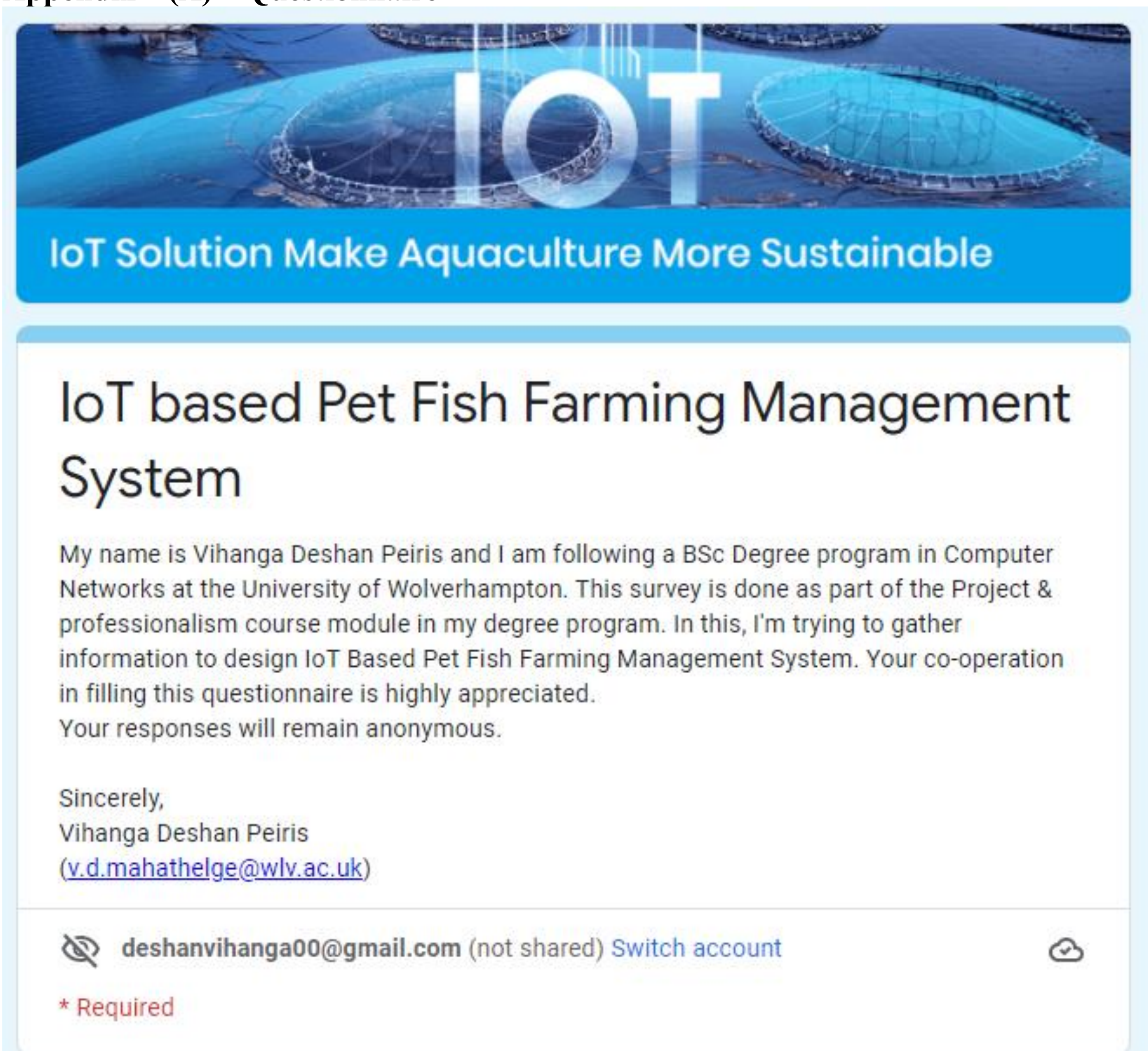
Compared to the findings I have found; the environmental condition of the fish tank is very important for the growth of the fish. It is also very important to increase the production of the farmer. Finally, and perhaps most importantly, this system enables the farmer to quickly and easily capture the tank's fish population. This is another factor that has a direct effect on fish growth. A general rule of thumb is that one inch of fish should be kept per two gallons of water. This, however, should be considered only to provide you with a general idea and feel for the limit. This can be done very quickly by the existing population counting system.

5.4 Future Work

The farmer must manually run the population counting system in this system, and only the population count is displayed. The system's future work will be to upgrade it so that it can automatically count and predict whether or not the fish in the tank will exceed the required size. In addition to the existing variables, there are other significant factors that could help fish live in a tank and be added to the system.

Appendix

Appendix – (A) – Questionnaire


IoT Solution Make Aquaculture More Sustainable

IoT based Pet Fish Farming Management System

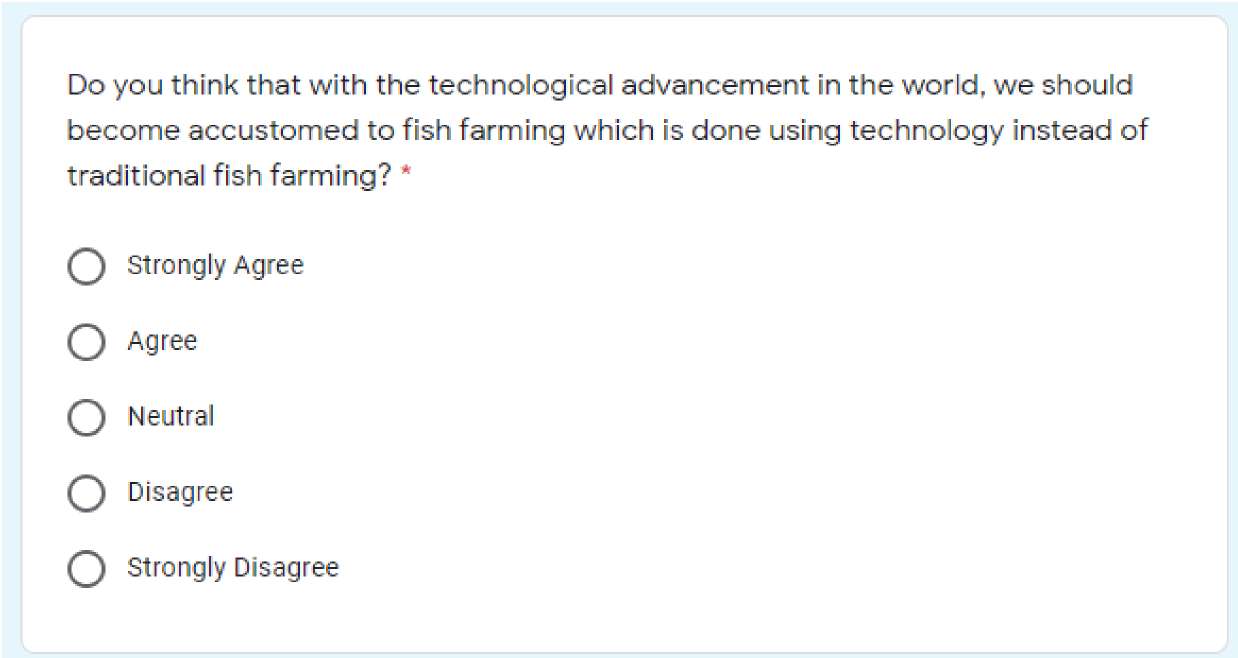
My name is Vihanga Deshan Peiris and I am following a BSc Degree program in Computer Networks at the University of Wolverhampton. This survey is done as part of the Project & professionalism course module in my degree program. In this, I'm trying to gather information to design IoT Based Pet Fish Farming Management System. Your co-operation in filling this questionnaire is highly appreciated.
Your responses will remain anonymous.

Sincerely,
Vihanga Deshan Peiris
(v.d.mahathelge@wlv.ac.uk)

deshanvihanga00@gmail.com (not shared) Switch account

* Required

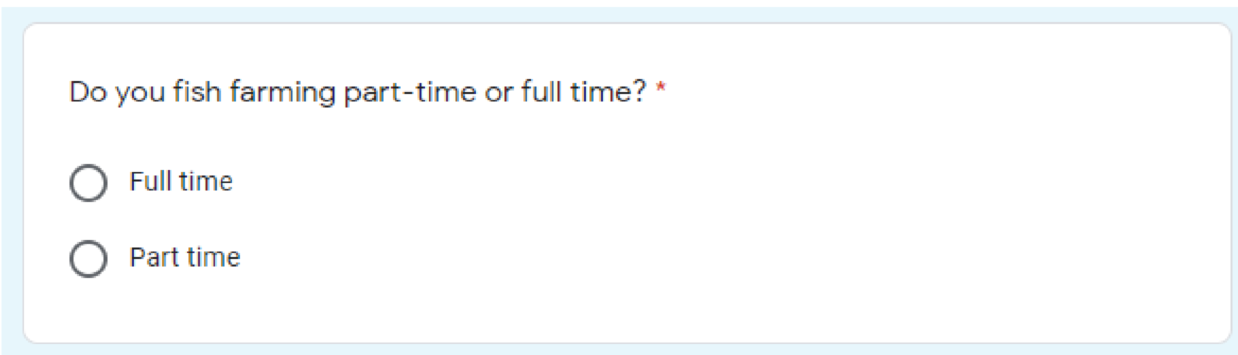
Figure 64 – questioner – 1



Do you think that with the technological advancement in the world, we should become accustomed to fish farming which is done using technology instead of traditional fish farming? *

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

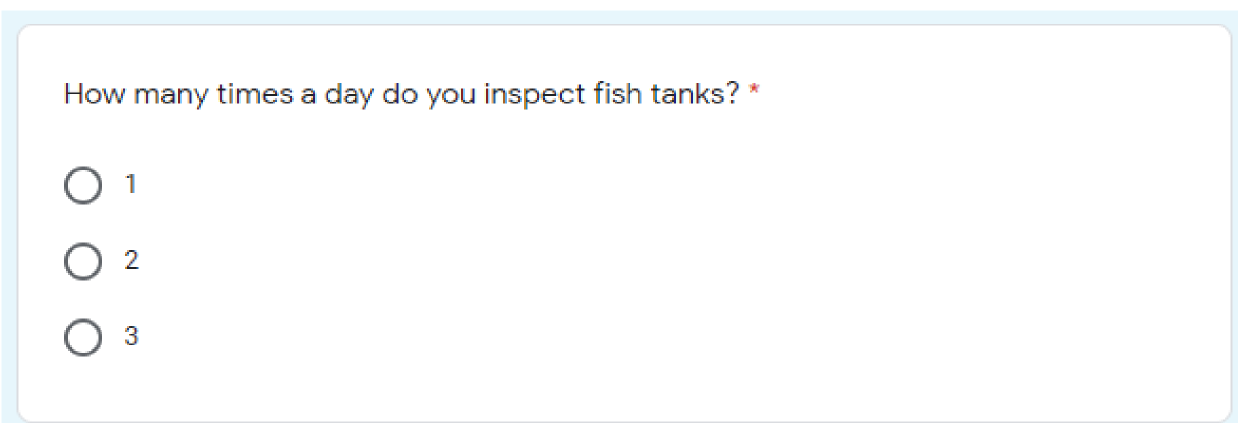
Figure 65 - questioner - 2



Do you fish farming part-time or full time? *

- Full time
- Part time

Figure 66 – questioner – 3



How many times a day do you inspect fish tanks? *

- 1
- 2
- 3

Figure 67 - questioner 4

What is your opinion on monitoring aquaculture using technology? *

- ○ Very Likely
- ○ Likely
- ○ Unlikely
- ○ Very Unlikely

Figure 68 – questioner - 5

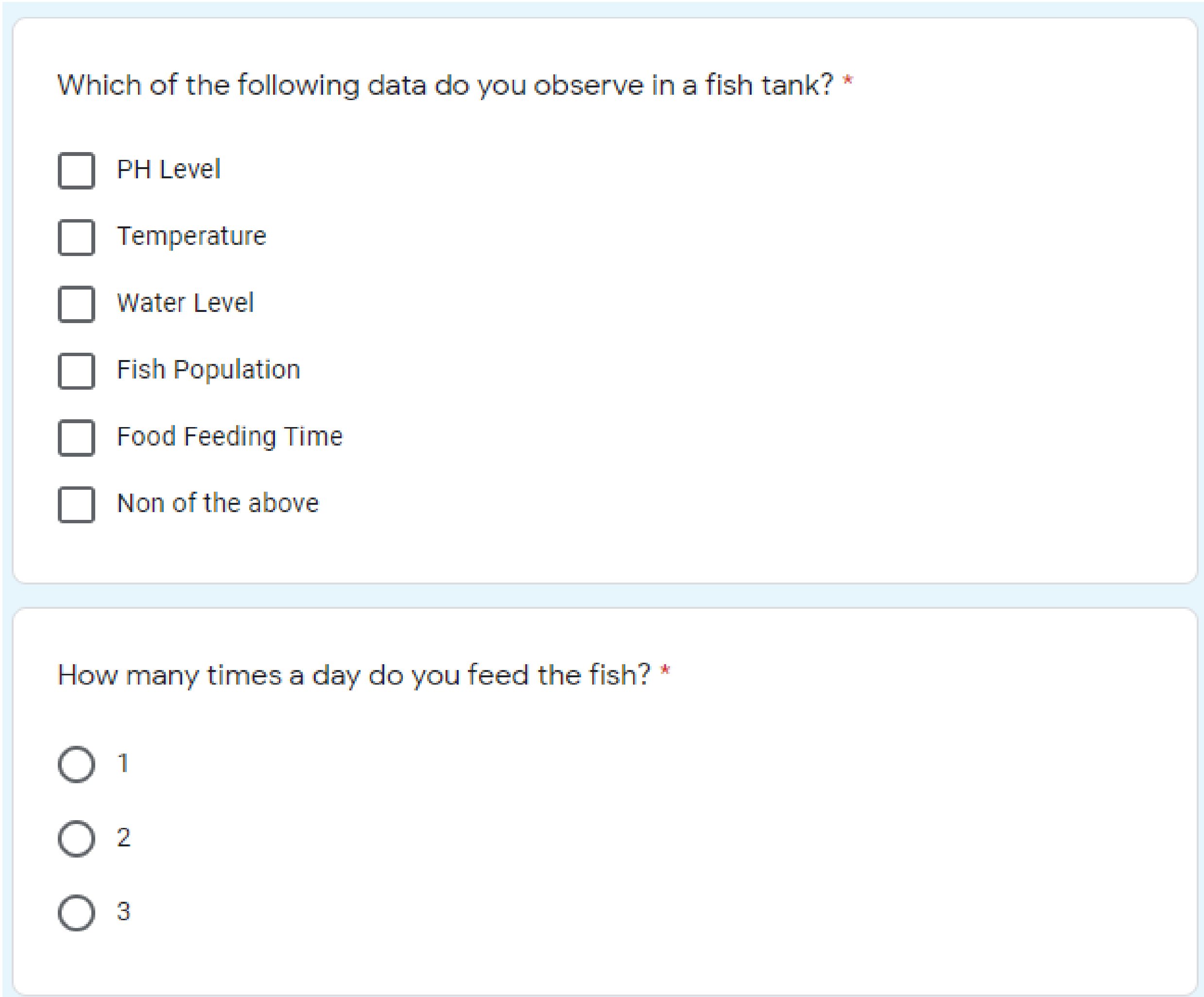
Do you keep track of the environmental conditions in the tanks? *

- ○ Yes
- ○ No

If you could use iot technology to get the data you need to monitor your daily fish tank from anywhere, what do you think? *

- ○ Very Likely
- ○ Likely
- ○ Unlikely
- ○ Very Unlikely

Figure 69 - questioner – 6



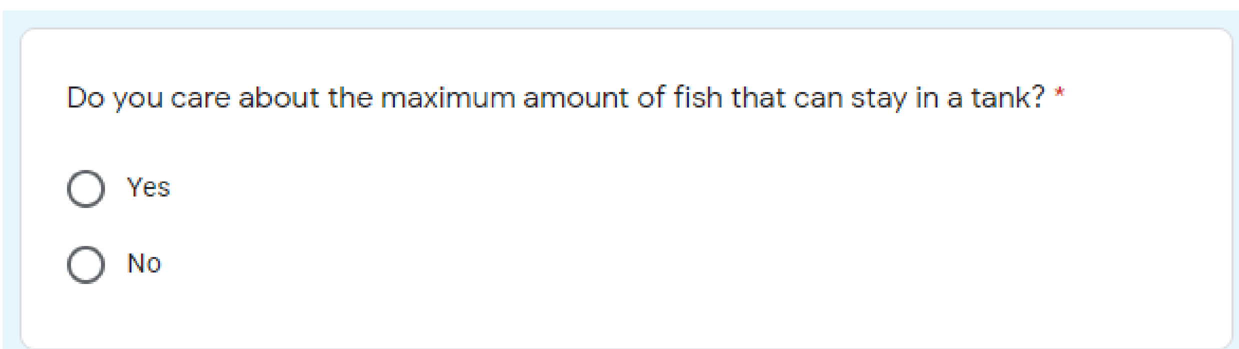
Which of the following data do you observe in a fish tank? *

- [ ] PH Level
- [ ] Temperature
- [ ] Water Level
- [ ] Fish Population
- [ ] Food Feeding Time
- [ ] Non of the above

How many times a day do you feed the fish? *

- ○ 1
- ○ 2
- ○ 3

Figure 70 - questioner 7



Do you care about the maximum amount of fish that can stay in a tank? *

- ○ Yes
- ○ No

Figure 71 - questioner 8

Do you agree that technology can reduce labor costs? *

- ○ Agree
- ○ Disagree

Finally, would you like to use a technology-based system to manage your aquaculture? *

- ○ Yes
- ○ No
- ○ Maybe

Figure 72 - questioner - 9

Appendix – (B) – Coding

Image Capturing System Coding

```php
<?php session_start(); /* Starts the session */

    /* Check Login form submitted */
    if(isset($_POST['Submit'])){
        /* Define username and associated password array */
        $logins = array('deshan' => '1234');

        /* Check and assign submitted Username and Password to new variable */
        $Username = isset($_POST['Username']) ? $_POST['Username'] : '';
        $Password = isset($_POST['Password']) ? $_POST['Password'] : '';

        /* Check Username and Password existence in defined array */
        if (isset($logins[$Username]) && $logins[$Username] == $Password){
            /* Success: Set session variables and redirect to Protected page  */
            $_SESSION['UserData']['Username']=$logins[$Username];
            header("location:gallery.php");
            exit;
        } else {
            /*Unsuccessful attempt: Set error message */
            $msg="<span style='color:red'>Invalid Login Details</span>";
        }
    }
?>
<!doctype html>
<html>
<head>
<title>Fish Image Capturing System</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css"/>
<style>
.header {
  overflow: hidden;
  background-color: #19302E;
  padding: 20px 10px;
}

.header a {
```

```css

.header a {
  float: left;
  color: white;
  text-align: center;
  padding: 12px;
  text-decoration: none;
  font-size: 18px;
  line-height: 25px;
  border-radius: 4px;
}

.header a.logo {
  font-size: 25px;
  font-weight: bold;
}

.header a:hover {
  background-color: #ddd;
  color: black;
}

.header a.active {
  background-color: dodgerblue;
  color: white;
}

.header-right {
  float: right;
}

.Table {
    font-family: verdana, Helvetica, sans-serif;
    font-size: 12px;
    color: #333;
    background-color: #E4E4E4;
}
.Table td {
```

C:\xampp\htdocs\j\login.php - Notepad++

```css
}
.Table td {
    background-color: #F8F8F8;
}

.container {
    padding: 16px;
    margin: auto;
    width: fit-content;
}

footer{
  background-color: #19302E;
  border-top: 1px solid #6EB981;
  font-size: 17px;
  padding: 15px 5px;
  color: #ffffff;
  text-align: center;
}

@media screen and (max-width: 500px) {
  .header a {
    float: none;
    display: block;
    text-align: left;
  }

.header-right {
    float: none;
  }
}

@media {
  .footer-links li {
    display: inline;
  }
}
```

PHP Hypertext Preprocessor file length : 5,127 lines : 247 Ln : 1 Col : 48 Sel : 0 | 0 Unix (LF) UTF-8 INS

C:\xampp\htdocs\j\login.php - Notepad++

```css
}

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
  }

.wrapper{
  width: 100%;
  background: #fff;
  border-radius: 5px;
  box-shadow: 0px 4px 10px 1px rgba(0,0,0,0.1);
}
.wrapper .title{
  height: 90px;
  background: #19302E;
  border-radius: 5px 5px 0 0;
  color: #fff;
  font-size: 30px;
  font-weight: 600;
  display: flex;
  align-items: center;
  justify-content: center;
}
.wrapper form{
  padding: 30px 25px 25px 25px;
}
.wrapper form .row{
  height: 45px;
  margin-bottom: 15px;
  position: relative;
}
.wrapper form .row input{
  height: 100%;
  width: 100%;
```

PHP Hypertext Preprocessor file length : 5,127 lines : 247 Ln : 1 Col : 48 Sel : 0 | 0 Unix (LF) UTF-8 INS

C:\xampp\htdocs\j\login.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

login.php | upload.php | gallery.php | script.php | logout.php

```
  position: relative;
}
.wrapper form .row input{
  height: 100%;
  width: 100%;
  outline: none;
  padding-left: 60px;
  border-radius: 5px;
  border: 1px solid lightgrey;
  font-size: 16px;
  transition: all 0.3s ease;
}
form .row input:focus{
  border-color: #16a085;
  box-shadow: inset 0px 0px 2px 2px rgba(26,188,156,0.25);
}
form .row input::placeholder{
  color: #999;
}
.wrapper form .row i{
  position: absolute;
  width: 47px;
  height: 100%;
  color: #fff;
  font-size: 18px;
  background: #19302E;
  border: 1px solid #16a085;
  border-radius: 5px 0 0 5px;
  display: flex;
  align-items: center;
  justify-content: center;
}
.wrapper form .pass{
  margin: -8px 0 20px 0;
}
.wrapper form .pass a{
  color: #19302E;
  font-size: 17px;
```

PHP Hypertext Preprocessor file | length : 5,127 lines : 247 | Ln : 1 Col : 48 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

C:\xampp\htdocs\j\login.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

login.php | upload.php | gallery.php | script.php | logout.php

```
}
.wrapper form .button input{
  color: #fff;
  font-size: 20px;
  font-weight: 500;
  padding-left: 0px;
  background: #19302E;
  border: 1px solid #16a085;
  cursor: pointer;
}
form .button input:hover{
  background: #12876f;
}
.wrapper form .signup-link{
  text-align: center;
  margin-top: 20px;
  font-size: 17px;
}
.wrapper form .signup-link a{
  color: #16a085;
  text-decoration: none;
}
form .signup-link a:hover{
  text-decoration: underline;
}

</style>
</head>
<body>
<div class="header">
  <a href="#" class="logo">Fish Image Capturing System</a>

</div>
<br>
<br>
<div class="container">
      <div class="wrapper">
        <div class="title"><span>Login Form</span></div>
```

PHP Hypertext Preprocessor file | length : 5,127 lines : 247 | Ln : 1 Col : 48 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

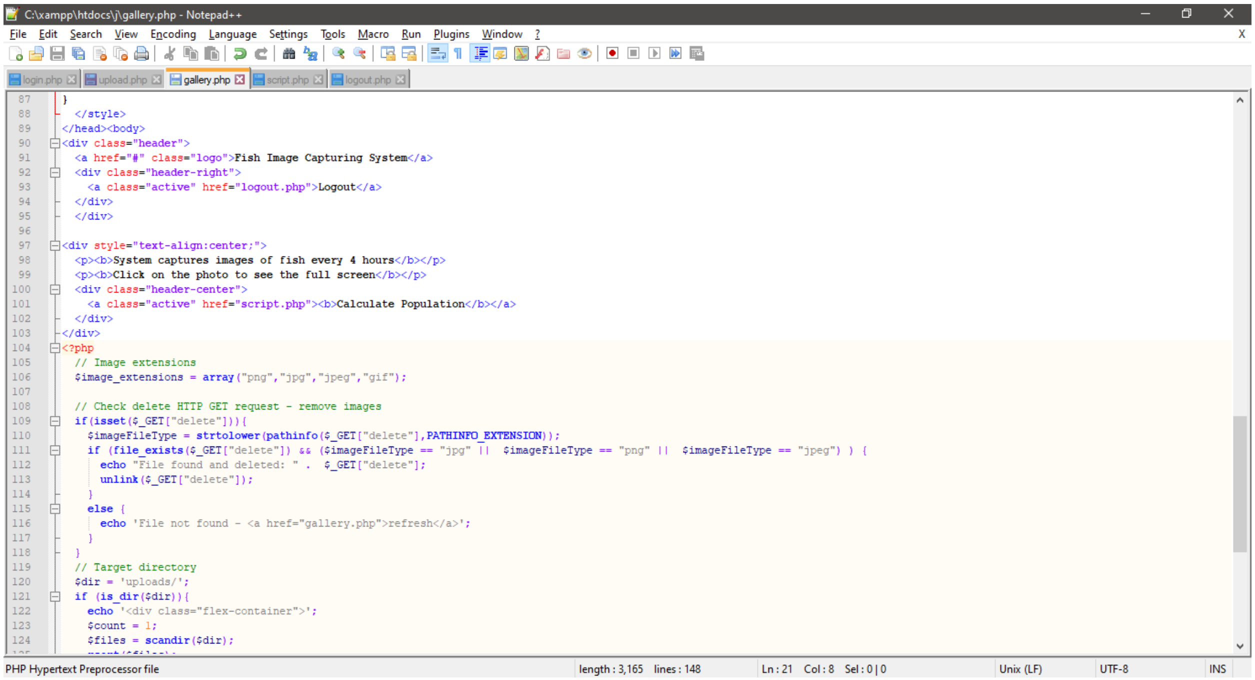
```
</style>
</head>
<body>
<div class="header">
  <a href="#" class="logo">Fish Image Capturing System</a>

</div>
<br>
<br>
<div class="container">
      <div class="wrapper">
        <div class="title"><span>Login Form</span></div>
        <form action="#" method="post" name="Login_Form">
          <div class="row">

            <i class="fas fa-user"></i>
            <input name="Username" type="text" placeholder="Username" required>
          </div>
          <div class="row">
            <i class="fas fa-lock"></i>
            <input name="Password" type="password" placeholder="Password" required>
          </div>

          <div class="row button">
            <input name="Submit" type="submit" value="Login">
          </div>

        </form>
        </div>
    </div>
</body>
<br>
<br>
<section id="footer">
        <footer>Fish Image Capturing System by Vihanga Deshan &copy; 2022 All Rights Reserved</footer>
    </section>

</html>
```

upload.php code

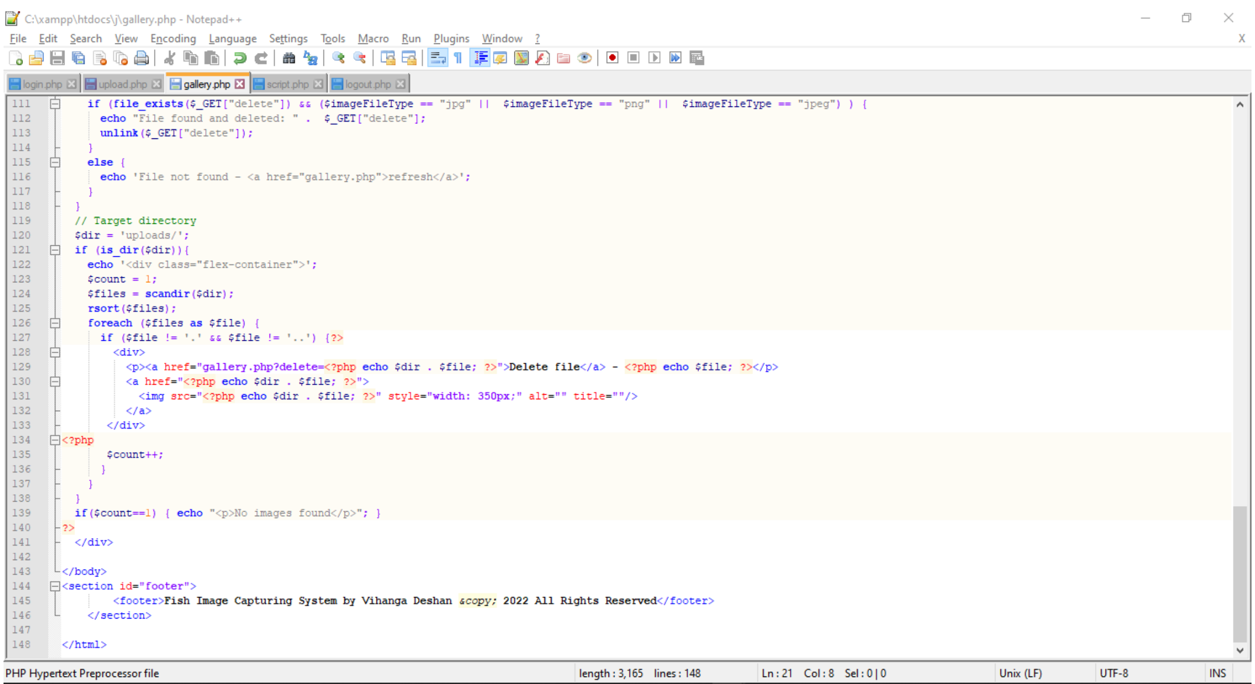
```
<?php

$target_dir = "uploads/";
$datum = mktime(date('H')+0, date('i'), date('s'), date('m'), date('d'), date('y'));
$target_file = $target_dir . date('Y.m.d_H:i:s_', $datum) . basename($_FILES["imageFile"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

if(isset($_POST["submit"])) {
  $check = getimagesize($_FILES["imageFile"]["tmp_name"]);
  if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
  }
  else {
    echo "File is not an image.";
    $uploadOk = 0;
  }
}

if (file_exists($target_file)) {
  echo "Sorry, file already exists.";
  $uploadOk = 0;
}

if ($_FILES["imageFile"]["size"] > 500000) {
  echo "Sorry, your file is too large.";
  $uploadOk = 0;
}

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
  echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
  $uploadOk = 0;
}

if ($uploadOk == 0) {
  echo "Sorry, your file was not uploaded.";
```

```
if ($uploadOk == 0) {
  echo "Sorry, your file was not uploaded.";
}
else {
  if (move_uploaded_file($_FILES["imageFile"]["tmp_name"], $target_file)) {
    echo "The file ". basename( $_FILES["imageFile"]["name"]). " has been uploaded.";
  }
  else {
    echo "Sorry, there was an error uploading your file.";
  }
}
?>
```

gallery.php



```
}
  </style>
</head><body>
<div class="header">
  <a href="#" class="logo">Fish Image Capturing System</a>
  <div class="header-right">
    <a class="active" href="logout.php">Logout</a>
  </div>
  </div>

<div style="text-align:center;">
  <p><b>System captures images of fish every 4 hours</b></p>
  <p><b>Click on the photo to see the full screen</b></p>
  <div class="header-center">
    <a class="active" href="script.php"><b>Calculate Population</b></a>
  </div>
</div>
<?php
  // Image extensions
  $image_extensions = array("png","jpg","jpeg","gif");

  // Check delete HTTP GET request - remove images
  if(isset($_GET["delete"])){
    $imageFileType = strtolower(pathinfo($_GET["delete"],PATHINFO_EXTENSION));
    if (file_exists($_GET["delete"]) && ($imageFileType == "jpg" ||  $imageFileType == "png" ||  $imageFileType == "jpeg") ) {
      echo "File found and deleted: " .  $_GET["delete"];
      unlink($_GET["delete"]);
    }
    else {
      echo 'File not found - <a href="gallery.php">refresh</a>';
    }
  }
  // Target directory
  $dir = 'uploads/';
  if (is_dir($dir)){
    echo '<div class="flex-container">';
    $count = 1;
    $files = scandir($dir);
```



```
    if (file_exists($_GET["delete"]) && ($imageFileType == "jpg" ||  $imageFileType == "png" ||  $imageFileType == "jpeg") ) {
      echo "File found and deleted: " .  $_GET["delete"];
      unlink($_GET["delete"]);
    }
    else {
      echo 'File not found - <a href="gallery.php">refresh</a>';
    }
  }
  // Target directory
  $dir = 'uploads/';
  if (is_dir($dir)){
    echo '<div class="flex-container">';
    $count = 1;
    $files = scandir($dir);
    rsort($files);
    foreach ($files as $file) {
      if ($file != '.' && $file != '..') {?>
        <div>
          <p><a href="gallery.php?delete=<?php echo $dir . $file; ?>">Delete file</a> - <?php echo $file; ?></p>
          <a href="<?php echo $dir . $file; ?>">
            <img src="<?php echo $dir . $file; ?>" style="width: 350px;" alt="" title=""/>
          </a>
       </div>
<?php
       $count++;
      }
    }
  }
  if($count==1) { echo "<p>No images found</p>"; }
?>
  </div>

</body>
<section id="footer">
        <footer>Fish Image Capturing System by Vihanga Deshan &copy; 2022 All Rights Reserved</footer>
    </section>

</html>
```

logout.php

```php
<?php session_start(); /* Starts the session */
session_destroy(); /* Destroy started session */
header("location:login.php");
exit;
?>
```

Image Processing Part Coding

code.py

```python
# Import libraries
import cv2
import sqlite3
import datetime
import csv
import os
import glob

conn = sqlite3.connect('fishpopulation.db')
c = conn.cursor()


def create_table():
    c.execute('CREATE TABLE IF NOT EXISTS FishPopulation (Date TIMESTAMP, Number REAL)')


currentDateTime = datetime.datetime.now()

folder_path = r'C:\xampp\htdocs\j\uploads'
file_type = '\*jpg'
files = glob.glob(folder_path + file_type)
max_file = max(files, key=os.path.getctime)

image = cv2.imread(max_file)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)

print("fish count in the image : ", len(cnt))

```

```python

def data_entry():
    date = currentDateTime
    number = len(cnt)

    c.execute("INSERT INTO FishPopulation (Date, Number) VALUES(?, ?)", (date, number))
    conn.commit()


print("Data Insert successfully")

create_table()
data_entry()


#def read_from_db():
#    c.execute('SELECT * FROM FishPopulation')
#    data = c.fetchall()
#    print("Recorded Data - ")
#    for row in data:
#        print(row)


print("Exporting data into CSV............")

cursor = conn.cursor()
cursor.execute("select * from FishPopulation")
with open("data.csv", "w") as csv_file:
    csv_writer = csv.writer(csv_file, delimiter="\t")
    csv_writer.writerow(['Column 1', 'Column 2'])
    csv_writer.writerows(cursor)

dirpath = os.getcwd() + "/data.csv"
print("Data exported Successfully into {}".format(dirpath))

c.close()
conn.close()

```

script.php

C:\xampp\htdocs\j\python\script.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

login.php upload.php gallery.php script.php logout.php

```php
<?php

$command = escapeshellcmd('python code.py');
$output = shell_exec($command);
echo $output;

?>
```

Node MCU Codding

water_quality_p1 | Arduino 1.8.13

File Edit Sketch Tools Help

water_quality_p1 §

```cpp
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define SensorPin A0  // the pH meter Analog output is connected with the Arduino’s Analog
int FloatSensor = 2;
int motor1 = 14; //D5
int motor2 = 12; //D6
int light = 13; //D7
int fan = 15; //D8

int buttonState = 1; //reads pushbutton status

unsigned long int avgValue;  //Store the average value of the sensor feedback
float b;
int buf[10], temp;
const int oneWireBus = 4;

char auth[] = "ITaapbhbY--eF-Lu5xp5bofvittirhmK";       // You should get Auth Token in the Blynk App.
char ssid[] = "SLT";                          // Your WiFi credentials.
char pass[] = "12345678";

```

water_quality_p1 | Arduino 1.8.13

File Edit Sketch Tools Help

water_quality_p1 §

```cpp
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

void setup()
{

  Serial.begin(115200); // Dubugging on hardware Serial 0
  Blynk.begin(auth, ssid, pass);
  sensors.begin();
  pinMode(FloatSensor, INPUT_PULLUP);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  pinMode(light, OUTPUT);
  pinMode(fan, OUTPUT);

}
void loop()
{
  Blynk.run();
  for (int i = 0; i < 10; i++) //Get 10 sample value from the sensor for smooth the value
  {
    buf[i] = analogRead(SensorPin);
    delay(10);
  }
```

water_quality_p1 | Arduino 1.8.13

```
  }
  for (int i = 0; i < 9; i++) //sort the analog from small to large
  {
    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++)                 //take the average value of 6 center sample
    avgValue += buf[i];
  float phValue = (float)avgValue * 5.0 / 1024 / 6; //convert the analog into millivolt
  phValue = 3.5 * phValue; //convert the millivolt into pH value

  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);

  buttonState = digitalRead(FloatSensor);
  //inside motor
  if ((phValue < 4) && ( buttonState == LOW)) {
```

water_quality_p1 | Arduino 1.8.13

```
    digitalWrite(motor1, HIGH);
  }
  else {
    digitalWrite(motor1, LOW);

  }
  //outside motor
  if ( buttonState == HIGH)) {
    digitalWrite(motor2, HIGH);
  }
  else {
    digitalWrite(motor2, LOW);

  }
  //fan
  if ( temperatureC > 32 ) {
  digitalWrite(fan , HIGH);
  }
  else {
    digitalWrite(fan , LOW);

  }
  //light
  if ( temperatureC < 25 ) {
  digitalWrite(light , HIGH);
  }
  else {
    digitalWrite(light , LOW);

  }

  Serial.print(temperatureC);
  Serial.print("°C");
  Serial.print("    pH:");
  Serial.print(phValue, 2);
  Serial.println(" ");


  Blynk.virtualWrite(V1, phValue);
  Blynk.virtualWrite(V2, temperatureC);
  delay(1000);
}
```

ESP32 CAM Module Coding

```
#include <Arduino.h>
#include <WiFi.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"

const char* ssid = "SLT_FIBRE";
const char* password = "abc12345";

String serverName = "192.168.1.2";
String serverPath = "/upload.php";

const int serverPort = 8089;

WiFiClient client;

// CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

const int timerInterval = 30000;    // time between each HTTP POST image
unsigned long previousMillis = 0;   // last time image was sent

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);

  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
```

```
  // init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_CIF;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }

```

```
void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= timerInterval) {
    sendPhoto();
    previousMillis = currentMillis;
  }
}

String sendPhoto() {
  String getAll;
  String getBody;

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
  }
```

```
  Serial.println("Connecting to server: " + serverName);

  if (client.connect(serverName.c_str(), serverPort)) {
    Serial.println("Connection successful!");
    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"imageFile\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--RandomNerdTutorials--\r\n";

    uint32_t imageLen = fb->len;
    uint32_t extraLen = head.length() + tail.length();
    uint32_t totalLen = imageLen + extraLen;

    client.println("POST " + serverPath + " HTTP/1.1");
    client.println("Host: " + serverName);
    client.println("Content-Length: " + String(totalLen));
    client.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
    client.println();
    client.print(head);

```

```
    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n=0; n<fbLen; n=n+1024) {
      if (n+1024 < fbLen) {
        client.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        client.write(fbBuf, remainder);
      }
    }
    client.print(tail);

    esp_camera_fb_return(fb);

    int timoutTimer = 10000;
    long startTimer = millis();
    boolean state = false;
```

```
    while ((startTimer + timoutTimer) > millis()) {
      Serial.print(".");
      delay(100);
      while (client.available()) {
        char c = client.read();
        if (c == '\n') {
          if (getAll.length()==0) { state=true; }
          getAll = "";
        }
        else if (c != '\r') { getAll += String(c); }
        if (state==true) { getBody += String(c); }
        startTimer = millis();
      }
      if (getBody.length()>0) { break; }
    }
    Serial.println();
    client.stop();
    Serial.println(getBody);
  }
```

```
  else {
    getBody = "Connection to " + serverName +  " failed.";
    Serial.println(getBody);
  }
  return getBody;
}
```

References

1. RAJA Subramanian, K., 2017. *IOT based Automation of Fish Farming*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/345895043_IOT_based_Automation_of_Fish_Farming> [Accessed 7 December 2021].

2. Janet, J., Balakrishnan, S. and Sheeba Rani, S., 2019. *IOT Based Fishery Management System*. [online] Ripublication.com. Available at: <https://www.ripublication.com/ijoo19/ijoov13n1_12.pdf> [Accessed 10 December 2021].

3. Ahmed, M., Rahaman, M., Rahman, M. and Abul Kashem, M., 2019. Analyzing the Quality of Water and Predicting the Suitability for Fish Farming based on IoT in the Context of Bangladesh. *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*,.

4. Idachaba, F., Olowoleni, J., Ibhaze, A. and Oni, O., 2017. *IoT Enabled Real-Time Fishpond Management System*. [online] Eprints.covenantuniversity.edu.ng. Available at: <http://eprints.covenantuniversity.edu.ng/10497/1/WCECS2017_pp42-46.pdf> [Accessed 16 December 2021].

5. Wang, C., Li, Z., Wang, T., Xu, X., Zhang, X. and Li, D., 2021. Intelligent fish farm—the future of aquaculture. *Aquaculture International*, 29(6), pp.2681-2711.

6. S, N. and A, N., 2022. *WATER MONITORING IOT SYSTEM FOR FISH FARMING PONDS*. [online] Stumejournals.com. Available at: <https://stumejournals.com/journals/i4/2018/2/77.full.pdf> [Accessed 4 December 2021].

7. Pasha Mohd Daud, A., Sulaiman, N., Mohamad Yusof, Y. and Kassim, M., 2020. An IoT-Based Smart Aquarium Monitoring System. *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*,.

8. Nasir, O. and Mumtazah, S., 2020. IOT-BASED MONITORING OF AQUACULTURE SYSTEM. *MATTER: International Journal of Science and Technology*, 6(1), pp.113-137.

9. Cruz, C., Fulla, A., Sorezo, P. and Balba, D., 2022. *AQUACULTURE MAINTENANCE AND MONITORING SYSTEM FOR COLD WATER AQUARIUM*. [online] Lpulaguna.edu.ph. Available at: <https://lpulaguna.edu.ph/wp-content/uploads/2018/12/10-AQUACULTURE-MAINTENANCE-AND.pdf> [Accessed 5 December 2021].

10. WSPglobal. 2022. *Automated Counter for Fish Populations Monitoring*. [online] Available at: <https://www.wsp.com/en-CA/services/automated-counter-for-fish-populations-monitoring> [Accessed 4 December 2021].

11. Han, J., Honda, N., Asada, A. and Shibata, K., 2009. Automated acoustic method for counting and sizing farmed fish during transfer using DIDSON. *Fisheries Science*, 75(6), pp.1359-1367.

12. Zhang, S., Yang, X., Wang, Y., Zhao, Z., Liu, J., Liu, Y., Sun, C. and Zhou, C., 2020. Automatic Fish Population Counting by Machine Vision and a Hybrid Deep Neural Network Model. *Animals*, 10(2), p.364.

13. Kori, S., Ayatti, S., Lalbeg, V. and Angadi, A., 2018. Smart Live Monitoring of Aquarium—An IoT Application. *Information and Communication Technology for Intelligent Systems*, pp.1-9.

14. PawTracks. 2022. *What Causes pH Levels to Rise in an Aquarium? | PawTracks*. [online] Available at: <https://www.pawtracks.com/other-animals/aquarium-ph-levels/> [Accessed 15 December 2021].

15. Algone. 2022. *Fish population - stocking the aquarium*. [online] Available at: <https://www.algone.com/aquarium-fish-population> [Accessed 15 December 2021].

16. The Spruce Pets. 2022. *What Temperature Is Correct for Community Aquarium Mix of Fish*. [online] Available at: <https://www.thesprucepets.com/aquarium-water-temperature-1381896> [Accessed 15 November 2021].

17. The Spruce Pets. 2022. *How Much (and How Often) Should I Feed My Aquarium Fish?*. [online] Available at: <https://www.thesprucepets.com/how-much-should-i-feed-my-fish-1378746> [Accessed 16 December 2021].