ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

UNIVERSITY COLLEGE OF ENGINEERING

AQUAFARM MONITORING SYSTEM USING IoT

Report submitted to

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

For the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

CH.B.NEERAJ SINGH

Reg. No. 178297601003

Under the esteemed guidance of

DR.D.LATHA, ASSISTANT PROFESSOR.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING

ADIKAVI NANNAYA UNIVERSITY

2017-2021

ADIKAVI NANNAYA UNIVERSITY

RAJAHMAHENDRAVARAM

UNIVERSITY COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"AQUAFARM MONITORING SYSTEM USING IoT"** is a bonafide work of **Mr.CH.B.NEERAJ SINGH**, Reg No.**178297601003** submitted to partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Information Technology by **ADIKAVI NANNAYA UNIVERSITY** during the period 2017-2021. This work is carried out in the Department of Computer Science and Engineering, University College of Engineering, Adikavi Nannaya University.

INTERNAL GUIDE **HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINER

DECLARATION

I, hereby declare that the project report entitled **AQUAFARM MONITORING SYSTEM USING IoT** done by me under the guidance of **Dr.D.LATHA**, Assistant Professor, Department of Computer Science, University College of Engineering, Adikavi Nannaya University, is submitted for the fulfilment of requirement for the award of the degree, Bachelors of Technology in Information Technology in the academic year 2017-2021. I assert the statements made and conclusions drawn are an outcome of my research work. I further certifythat I have followed the guidelines provided by the university in writing the report. Whenever I have used materials from other sources, I have given due credit to them in the text of the report and giving their details in the references.

Signature of the Student

ACKNOWLEDGEMENT

As I present my project on "**AQUAFARM MONITORING SYSTEM USING IoT**", I take this opportunity to offer my sincere thanks to all those without whose guidance this project mighthave remained a dream for me.

I express my deepest gratitude and thanks to **Dr. D.LATHA** whose guidance andideas channeled my conscientious endeavors towards the project. I have been fortunate enough that my guide gave me the freedom, support and whole hearted co-ordination for the completion of my project.

I profusely thank **Dr.D.LATHA**, Assistant Professor, Department of Computer Science, University College of Engineering for all the encouragement and support.

I also thank **Dr.P.VENKATESWARA RAO** , Head of the Department, Department of Computer Science and Engineering for the guidance throughout the academic.

A great deal of thanks to the entire Review panel Faculty of University College Of Engineering, Adikavi Nannaya University for their support and value added suggestions.

CH.B.NEERAJ SINGH

Reg. no: 178297601003

ABSTRACT

Aquaculture, also known as Aqua farming, is the farming of aquatic organisms such as fishes, shrimps and crabs. Water quality is the major factor that influences the growth of fish and other aquatic organisms. The water quality is measured in terms of parameters such as pH value, water levels, amount of dissolved oxygen, temperature and levels of turbidity that influences the growth and health conditions of the fish and other aquatic organisms.

Hence continuous monitoring of these parameters is essential. Through monitoring we can reduce the risks in aqua farming. My proposed work supports the remote monitoring of the aqua farming system based on Internet of Things (IoT) for real time monitor and control of a aqua farming system.

By integrating the system with the server, we can view the data and alert the user for any change in the required specifications periodically and store it for future analysis. The alert is shown to the user if any anomaly is detected so that he can initiate control activities to rectify the anomalies.

LIST OF CONTENTS

CHAPTER-1

INTRODUTION

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives.

Aquaculture is one of the thriving area in many countries in the world since demand for fish expanding day by day. According to The United Nations Food and Agriculture Organization (UNFAO) "2012 State of World Fisheries and Aquaculture ", Worldwide yearly production of fishery items add up to around 128 million tons. The animal protein intake per individual is about 15% and increases the human reliance on fishery resources. The average consumption of fish products is 19 to 20 kg per person per year today and will be 16.7 kg per year in 2030 according to UNFAO. Production of fisheries, advancement and future food needs are firmly related. Aquaculture comprises of the set of exercises, information and techniques for the rearing of aquatic plants and a few animal groups. This activity has an awesome significance in financial improvement and food production. Commercial aquaculture is confronting numerous issues because of sudden climatic vacillation leading to changes in water quality parameters.

1.1 Problem Statement:

1.1.1 Existing System:

In Existing System Aqua farmers are relying upon manual testing for knowing the condition of the various parameters of the water. But this manual testing (once in a week) is time consuming and also give inappropriate results as parameters for measuring water quality changes continuously with environment.

1.1.2 Proposed System:

In the proposed System, we are adding the modern technology to aquaculture to overcome this problem. So we have to be more selective in choosing the appropriate technologies for this kind of advancement. Nodemcu esp32 is used in our system for data processing which has an inbuilt Wi-Fi module and 000webhost as its server , storing device .Using these and performing some analysis on the water quality parameters make our work unique.

CHAPTER-2

SYSTEM ANALYSIS

2.1 Functional Requirements:

The functional requirements for a system describe what the system should do. Those requirements depend on the type of software being developed, the expected users of the software. These are statements of services the system should provide, how the system should react to inputs and how the system should behave in particular situation.

1. Temperature and Humidity should be monitored.
2. Ph and DO should be monitored.
3. Turbidity should be monitored.
4. Water level should be monitored.
5. IoT data should be sent to server.
6. The monitored values should be shown to the user.

2.2 Non Functional Requirements:

Non-functional requirements are actually the requirements that are not directly concerned with the specified function delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Some of the nonfunctional requirements related with this system are hereby below:

1. **Scalability:** The application can be extended to support other Aqua Farms also.

2. **Reliability:** The ability of the system to behave consistently in a user acceptable manner.
3. **Usability:** This project has faster response.
4. **Security:** This project has a secure connection between esp32 and web server using SSL.

2.3 Software and Hardware Requirements

2.3.1 Hardware Requirements

- Nodemcu Esp32.
- DHT11 or Temperature sensor.
- Turbidity sensor.
- PH sensor.
- DO sensor.
- Water level sensor.
- Resistors.
- Jumper Wires.
- Bread board.
- Web server with domain.
- Port c adaptor.
- Working System with RAM > 3GB

2.3.2 Software Requirements:

- Programming Language- C.
- Web Technologies- PHP, CSS, JS.
- Markup Languages- HTML.
- Database –MySQL.
- Software- Arduino IDE.

2.4 Feasibility Study:

A key part of the preliminary investigation that reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time factors. The purpose of the study is to determine if the systems request should proceed further.

2.4.1 Economical Feasibility:

This software is economic from Aqua farmer's point of view. It is cost effective as involvement of middleman has been eliminated. It is time effective because the farmer opens the portal and controls the devices or the system does itself depend on the mode of operation. Hence he will have control over the farm even he is not physically available. He can easily find the status of fish health by uploading the image in the website.

2.4.2 Technical Feasibility:

The technical requirement for the system is economic and it does not use any other additional Hardware and software.

2.4.3 Behavioral Feasibility:

The system working is quite easy to use. The Aqua farmer requires no special training for operating the system

CHAPTER-3

SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture, modules, and components, the different interfaces of those components, and the data that goes through that system. It is meant to satisfy the specific needs and requirements of a business or organization through the engineering of a coherent and well-running system

System design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process systematic where it takes into account all related variables of the system that needs to be created from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system, Systems design then overlaps with systems analysis, systems engineering, and systems architecture.

The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory operation.

3.1 System Components and Architecture:

The system consists of quality monitoring devices to measure the quality of aqua farm and also web application in which user can see the values of updated quality measures which takes data the esp32 periodically.

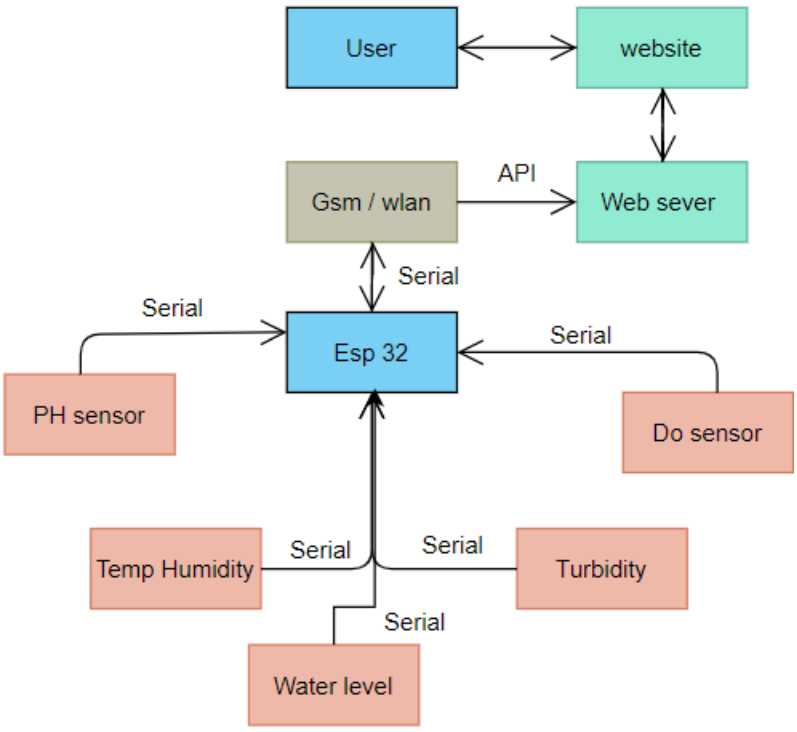

Figure 3.1.1 System Design

3.1.1 System Components

System modules are the components of a project/system in which each component has a specific job to do and also the components interact between them. Sometimes jobs are inter-dependent and each module has a specific set of actors which perform tasks.The components used in the System are:

- NODEMCU esp Wroom-32.
- Turbidiy sensor.
- Dth11 sensor or Temperature sensor
- Ph sensor.
- Do sensor.
- Water level sensor.

3.1.1.1 NODEMCU esp Wroom 32:

- NODEMCU esp wroom 32 is the microcontroller, It is the successor of esp 8266 with lots of new features loaded. It combines WiFi and Bluetooth wireless capabilities . NODEMCU esp 32 targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding. Esp 32 has dual core i.e. processors. It also has wide variety of peripherals available, like: capacitive touch, ADCs, DACs, UART, SPI, I2C and much more. It runs 32 bit programs. The clock frequency can go up to 240MHz and it has a 512 kB RAM.

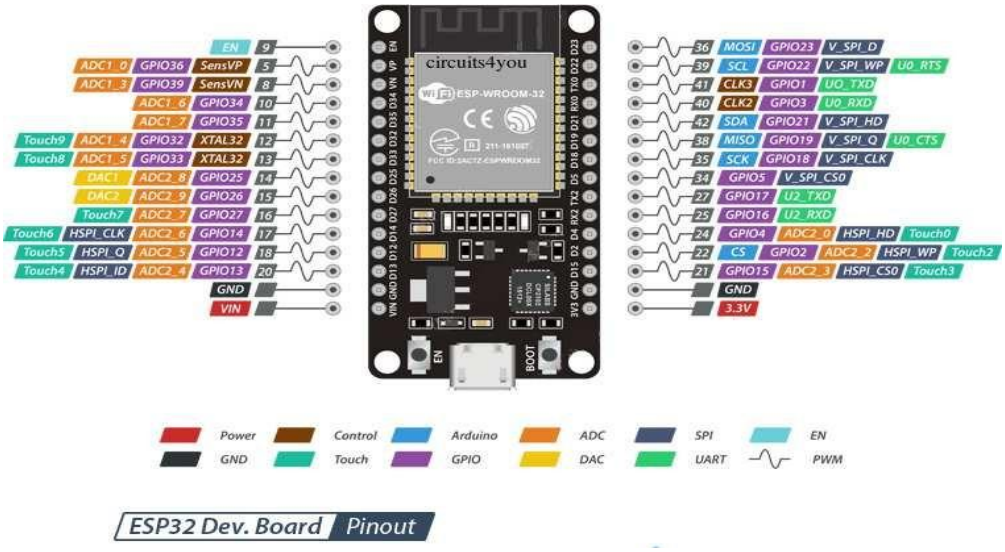

Figure 3.1.1.1.1 Esp32 Wroom

- We can use different programming environments in esp32 .
  - Arduino IDE
  - Espressif IDF (IoT Development Framework)
  - Micro Python
  - Javascript
  - LUA…….

- **Why Esp 32 ?????**

  Esp 32 have the dual core with a main advantage of more processing speed than other board of same prince range .Not only processing but also contains more GPIO pins which gives more possibility of input and output devices to connect. It contains the wifi and Bluetooth which gives more options in communication. Other difference are shown in the below table.

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

Figure 3.1.1.1.2 Differences between Esp 32 and other microcontrollers

- The Peripheral Feature of esp 32
  - 18 Analog-to-Digital Converter (ADC) channels
  - 10 Capacitive sensing GPIOs
  - 3 UART interfaces
  - 3 SPI interfaces
  - 2 I2C interfaces
  - 16 PWM output channels
  - 2 Digital-to-Analog Converters (DAC)
  - 2 I2S interfaces

ESP32 Configuration

Pin Category	Pin Name	Details
Power	**Micro-USB, 3.3V, 5V, GND**	**Micro-USB:** ESP32 can be powered through USB port **5V:** Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board. **3.3V:** Regulated 3.3V can be supplied to this pin to power the board. **GND:** Ground pins.
Enable	**En**	The pin and the button resets the microcontroller.
Analog Pins	**ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9**	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC
DAC pins	**DAC1 and DAC2**	Used for Digital to analog Conversion
Input/Output	**GPIO0 to GPIO39**	Totally 39 GPIO pins, can be used

Pins		as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	**T0 to T9**	These 10 pins can be used as touch pins normally used for capacitive pads
RTC GPIO pins	**RTCIO0 to RTCIO17**	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be used to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM through the software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.

HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	**AREF**	To provide a reference voltage for input voltage.

Technical Specifications

Microprocessor	Tensilica Xtensa LX6
Maximum Operating Frequency	240MHz
Operating Voltage	3.3V
Analog Input Pins	12-bit, 18 Channel
DAC Pins	8-bit, 2 Channel
Digital I/O Pins	39 (of which 34 is normal GPIO pin)
DC Current on I/O Pins	40 mA

DC Current on 3.3V Pin	50 mA
SRAM	520 KB
Communication	SPI(4), I2C(2), I2S(2), CAN, UART(3)
Wi-Fi	802.11 b/g/n
Bluetooth	V4.2 – Supports BLE and Classic Bluetooth

3.1.1.2 DHT 11

DHT-11 is a temperature and humidity sensor. The DHT11 sensor consists of two main components – one is Humidity sensing component (substrate) and the other one is NTC temperature sensor (or Thermistor). The thermistor is actually a variable resistor that changes its resistance with change in temperature. They both sense the temperature and humidity of an area and give the output to the IC (which is placed on back side of sensor). The sensor has four pins – VCC, Ground, data Out and NC. The VCC and Ground pins are connected to the common VCC and Ground respectively. The Data Out pin of the sensor is connected to GPIO pin of the esp 32.

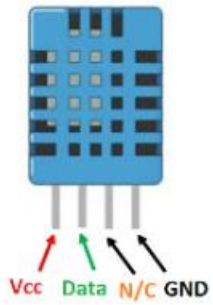

Figure 3.1.1.2.1 DHT 11 sensor

Internal Component will look like the below Diagram.

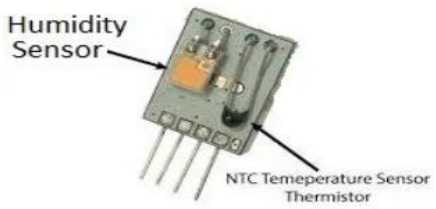

Figure 3.1.1.2.2 Internal parts of DHT 11 sensor

3.1.1.3 Water level sensor:

This sensor can be used to measure the water level, monitor a sump pit, detect rainfall or detect leakage. The sensor has a series of ten exposed copper traces, five of which are power traces and five are sense traces.These traces are interlaced so that there is one sense trace between every two power traces. Usually these traces are not connected but are bridged by water when submerged. The series of exposed parallel conductors, together acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water level. The resistance is inversely proportional to the height of the water

S-signal pin is an analog output that will be connected to one of the input pin of esp32.

+(vcc) pin supplies power for the sensor(3.3-5V).

-(GND) pin is a ground connection.

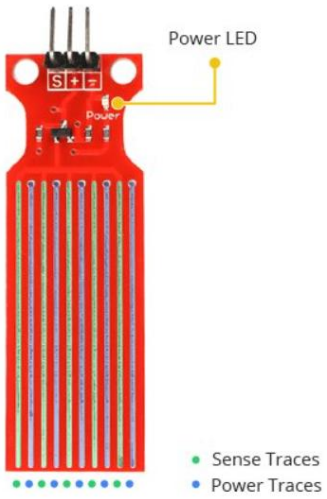

Figure 3.1.1.3.1 Water Level sensor

3.1.1.4 Turbidity Sensor

DFRobot's turbidity sensor contains two parts: the sensor to be submerged and the controller board. It describes the scattering of light projected into a liquid with suspended particles. Basically, the greater the number of particles, the more the light will be scattered. The turbidity sensor contains a light transmitter and receiver. At clear waters, light scattering is minimum and so the light receiver receives the most amount of light. As turbidity of the water increases, the light receiver receives less and less light. The sensor triggers when the light received is below a certain threshold.The Turbidity is measured in the NTU units.
The formula for the NTU units from the input value(volt) is given by

ntu = -1120.4*square(volt)+5742.3*volt-4352.9

VCC, GND and SIGNAL pins Are used for supplies power for the sensor (3.3-5V) ,Ground and sending data to esp32 GPIO pin .

Figure 3.1.1.4.1 Turbidity sensor.

3.1.1.5 PH sensor

Potentiometric PH measurement is the measurement of a potential difference using so-called glass electrodes or non glass electrodes. In the case of glass electrodes that pH sensitive element is a glass bulb that is fused to the end of a glass tube. The electrode is filled with a neutral potassium chloride solution buffered at pH 7 and contains a silver chloride wire that forms the electrical connection. when the pH sensor is immersed into this acid. The hydrogen ions are able to penetrate the boundary area of the glass membrane, so called gel layer. The considerably larger chloride ions remain in the solution the result is a charge separation. The same process occurs on the inside of the sensor with the neutral solution buffered at pH 7 that has a constant concentration of hydrogen ions. The resulting flow of current is in direct proportion to the pH value of the medium.

We convert the input value into PH using the following formula.

```
milivolt=(float) Value*5.0/1024/6; //convert the analog into millivolt
phValue=3.5*milivolt;
```

+(VCC) for the supply to the PH sensor ,-(GND) for ground and S(singal) for input the value to the esp32.

Figure 3.1.1.5.1 PH sensor.

3.1.1.6 DO sensor

This dissolved oxygen sensor is used to measure the dissolved oxygen in water, to reflect the water quality. Low dissolved oxygen in the water will lead to difficulty in breathing for aquatic organisms, which may threaten their lives. The filling solution and membrane cap are replaceable, leading to the low maintenance cost. Dissolved oxygen concentration levels may be expressed as milligrams per liter (mg/L) or parts per million (ppm)The sensor tip is covered by a membrane cap where a solution is filled. The oxygen-permeable membrane in the membrane cap is sensitive and fragile. The permeable membrane allows oxygen from the sample water to diffuse into the sensor, where it is reduced at the cathode. This chemical reaction produces an electrical signal, which travels from the cathode to the anode and then into the dissolved oxygen measuring instrument

+(VCC) for the supply to the PH sensor ,-(GND) for ground and S(singal) for input the value to the esp32.

Figure 3.1.1.6.1 Turbidity sensor.

3.1.2 Circuit diagram

After joining all the components into system the circuit will look like the following

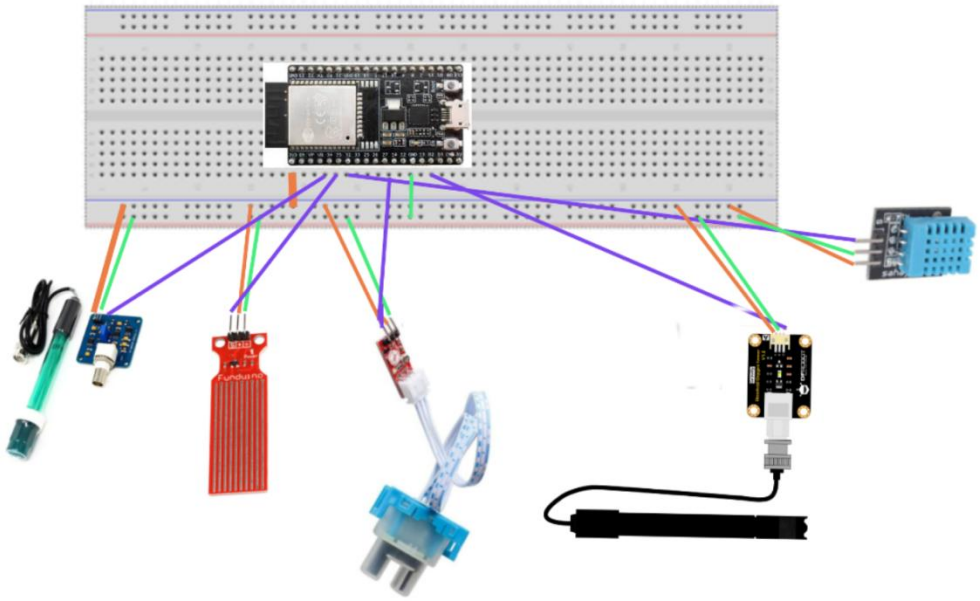

Figure 3.1.2.1 Circuit diagram.

As the circuit shows the esp 32 is connected to the sensors wih the help of jumper wires and using breadboard.

3.1.3 Architecture

Architecture of internet of Things contains basically 3 layers:

- Perception layer.
- Network layer.
- Application layer.

3.1.3.1 Perception Layer:

It is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

In this layer all the sensors PH ,DO , DHT 11 ,Water level sensors sense the Environment values that they are placed in.

3.1.3.2 Network Layer:

This layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

In this layer we uses Esp32 , wifi router manages the data and transmitted .

3.1.3.3 Application Layer:

This layer is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed.

In this layer 000webhost server and website is used as medium between user and system.

3.2 UML Diagrams

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. UML created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Rose Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

3.2.1 Use case Diagram

- A use case diagram is usually simple. It does not show the detail of the use cases:
- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

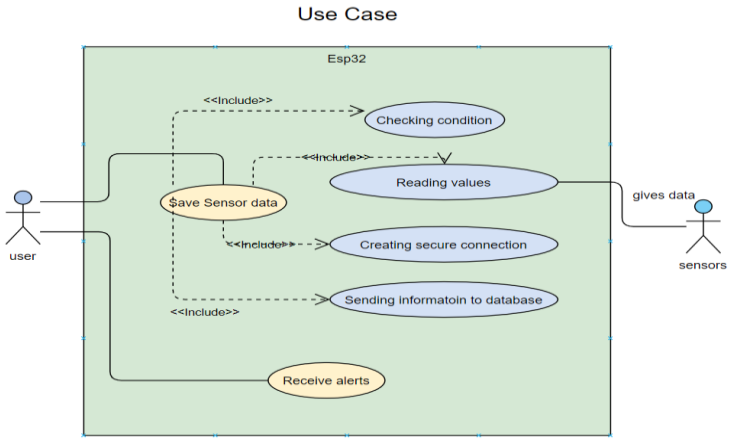

Figure 3.2.1.1 Use case diagram

3.2.2 Sequence Diagrams

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

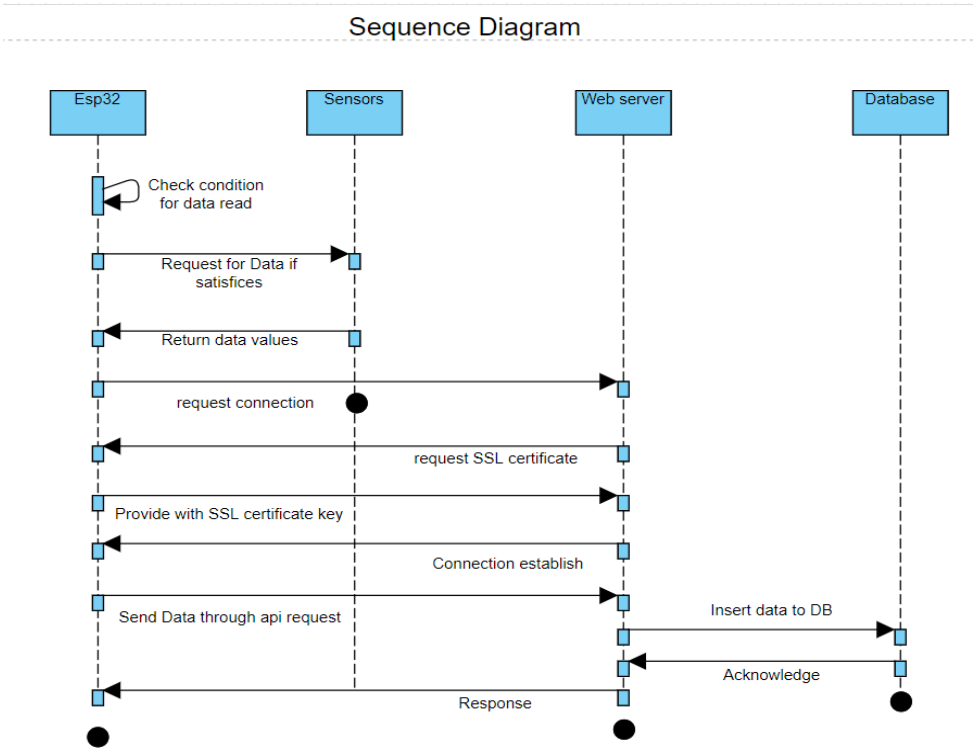

Figure 3.2.2.1 Sequence diagram

3.2.3 Activity Diagram

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

- Activities
- Association
- Conditions
- Constraints

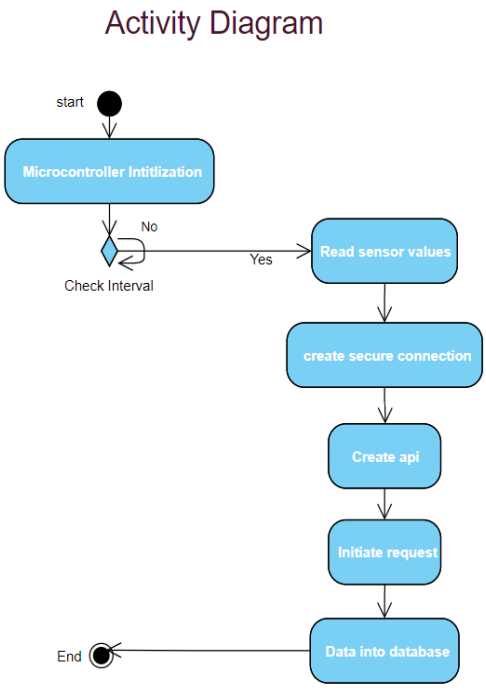

Figure 3.2.3.1 Activity diagram

3.3 Database design

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The following is the database table that is used to design my project module IoT based monitoring system for Aqua farms. Whenever the api request is get from the esp32 to server the database values of sensor get updated. The database design is shown in the below diagram.

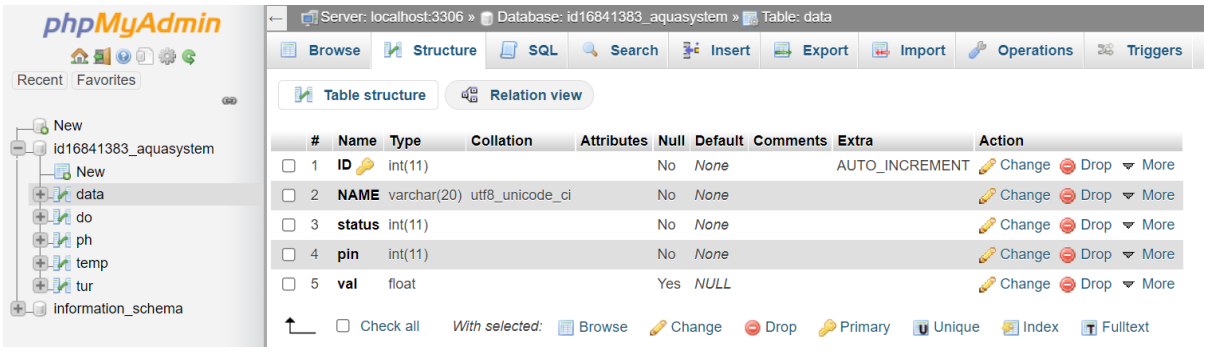

Figure 3.3.1 Database schema.

CHAPTER-4
IMPLEMENTATION

In this project the implementation take place in different phases as shown below in sequence order.

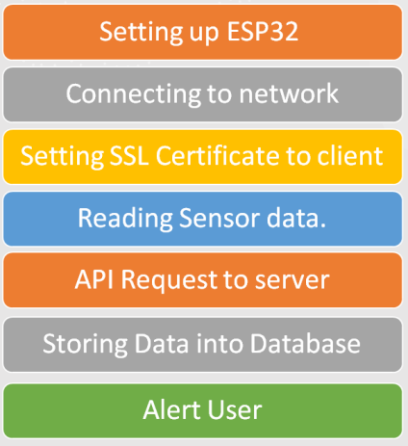

Figure 4.1. Phases of implementation

Let's see on by one

4.1 Setting up esp 32

In order to write a code in esp32 we need an environment, so, for that we are using Arduino IDE and the language we use is C language , let's see about them.

4.1.1. Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Esp 32 board. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the

programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the ESP 32 board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

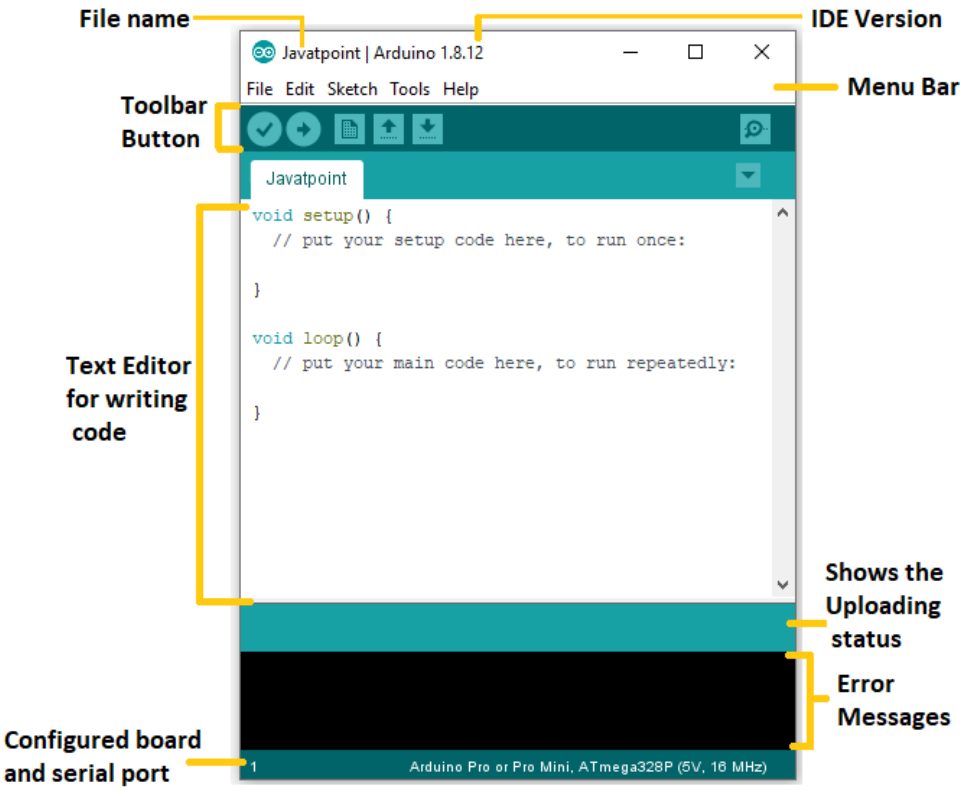

Figure 4.1.1.1 IDE

Toolbar Button

The icons displayed on the toolbar are **New, Open, Save, Upload,** and **Verify**.

It is shown below:

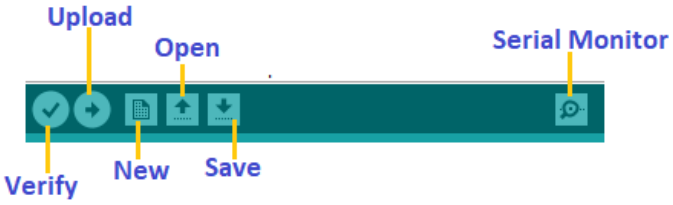

Figure 4.1.1.2 Tool bar

Serial Monitor

The serial monitor button is present on the right corner of the toolbar. It opens the serial monitor.

It is shown below:

Figure 4.1.1.3 Serial monitor

Manage Libraries...

It shows the updated list of all the installed libraries. We can also use this option to install a new library into the Arduino IDE.

And many more options are available in the Arduino IDE for developers.

4.1.2 Introduction to C language

C is a general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system. By design, C provides constructs that map efficiently to typical machine instructions. 25 C is an imperative procedural language. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code. Arduino programs are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution. The version of C can be used.

4.1.2. Importing Libraries

After setting the environment we need to select the port and device as nodemcu 32s

After the port and device is selected, The libraries needed for the project are imported.

```
#include <DHT.h>
#include <WiFiClientSecure.h>
#include<cmath>
```

Figure 4.1.2.1 Importing libraries

In which the DHT.h library is used to manage the DHT sensors ,which is written by Adafruit Industries.

The wifiClientSecure.h is used for secure connections using TLS (SSL). It inherits from WiFiClient and thus implements a superset of that class' interface. There are three ways to establish a secure connection using the WiFiClientSecure class: using a root certificate authority (CA) cert, using a root CA cert plus a client cert and key, and using a pre-shared key (PSK).

The cmath.h is used for the Mathematical functions that are used in the Project.

4.1.3 Setting pin modes

Pinmodes() is used for Configuring the specified pin to behave either as an input or an output. As of Arduino 1.0.1, it is possible to enable the internal pull up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull ups.

Syntax: pinMode(pin, mode);

```
pinMode(33,INPUT);
pinMode(35,INPUT);
pinMode(32,INPUT);
pinMode(26,INPUT);
pinMode(27,INPUT);
```

Figure 4.1.3.1 Setting pin mode

We have taken the five pins for the five sensors and made them pinMode as INPUT so that we can read inputs from them to esp32.

4.1.4. Initialize variables

In this phase we are taking the variables of float and assigning 0 to them. This variables are used to store the data that is retrieved from sensor or give alias name for the pins to remember.

```
//long countertimeon=300000;
long countertimewhenactoron=30000;
//long countertimewhenactoroff=5;
long countertimeoff=40000;
float dthtemp=0;
float dthhum=0;
float phval=0;
float turbval=0;
float doval=0;
float waterlev=0;
```

Figure 4.1.4.1 Initialize variables.

4.1.5 Setting Serial communication rate

The serial communication is a simple scheme that uses the UART (Universal Asynchronous Receiver/Transmitter) on the Microcontroller. Every message sent on the UART is in the form of 8 bits or 1 byte.

The serial.begin( ) *sets the baud rate for serial data* communication.
The **baud** rate signifies the data rate in bits per second.

We used speed of **115200 bps (bits per second**). We can specify other baud rates as well, such as 4800, 14400, 38400, 28800, etc.

Syntax: serial.begin(speed) or serial.begin(speed,config)

```
Serial.begin(115200);
```

Figure 4.1.5.1 Setting serial communication rate

4.2. Connecting to network:

The esp 32 can communicate with other device in many ways like wifi ,GSM , Bluetooth etc.

We are using Wifi as a medium to communicate between esp32 and server.

So we need to connect the wifi to 0router, for that fist we need to provide the ssid and password for authentication of router.

```
const char* ssid     = "realm";     // your network SSID (name of wifi network)
const char* password = "12345678"; // your network password
```

Figure 4.2.1 Setting credentials

After providing the ssid and password we request the router to connect.

```
WiFi.begin(ssid, password);

// attempt to connect to Wifi network:
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  // wait 1 second for re-trying
  delay(1000);
}

Serial.print("Connected to ");
Serial.println(ssid);
```

Figure 4.2.2 Initiate request

4.3 Setting SSL to client:

SSL certificates are what enable websites to move from HTTP to HTTPS, which is more secure. An SSL certificate is a data file hosted in a website's origin server. SSL certificates make SSL/TLS encryption possible, and they contain the website's public key and the website's identity, along with related information. Devices attempting to communicate with the origin server will reference this file to obtain the public key and verify the server's identity. The private key is kept secret and secure.

This SSL used for encryption secure and authentication of connection.

Steps to access the SSL certificate from the website is

- Visit website of server, you want connect.
- Go to certificate option.
- Go to certificate path.
- Select the Root directory from it.
- Click on view certificate option.
- Go to detail option
- Select copy file option.
- Click next button and save it in system.

After getting the certificate open in notepad and Use this key in program .as shown bellow.

```
const char* test_root_ca= \
    "-----BEGIN CERTIFICATE-----\n" \
"MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh\n" \
"MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3\n" \
"d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD\n" \
"QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT\n" \
"MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j\n" \
"b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG\n" \
"9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB\n" \
"CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97\n" \
"nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt\n" \
"43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P\n" \
"T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4\n" \
"gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO\n" \
"BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR\n" \
"TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw\n" \
"DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr\n" \
"hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg\n" \
"06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF\n" \
"PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls\n" \
"YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk\n" \
"CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=\n" \
"-----END CERTIFICATE-----";
```

Figure 4.3.1 SSL Certificate

In order to communicate with private server we are making the esp32 as a client (we can also make server) .And assigning the SSL certificate to client as shown below.

```
client.setCACert(test_root_ca);
//client.setCertificate(test_client_key); // for client verification
//client.setPrivateKey(test_client_cert); // for client verification

Serial.println("\nStarting connection to server...");
if (!client.connect(server, 443))
  Serial.println("Connection failed!");
else {
  Serial.println("Connected to server!");
  // Make a HTTP request:

}
```

Figure 4.3.2 Setting certificate to client

We are setting certificate to client using setCACert() and start making request to server using port number 443.

4.4 Reading values

4.4.1 Condition check

The esp32 uses the loop to repeat the process continuously, But reading values in every loop is not necessary .In order to read sensor data from sensors in optimized way .We can optimize them using 3 conditions.

- When the system starts for first time we will read the values
- When the System actors are on we will read data for every 5 mins.
- When all the actors are off then for every 3 hours we are reading the data.

Lets see the condition in the code below:

```
if((dthtemp==0 && dthhum==0 && phval==0 && doval==0 && waterlevel==0)||
((au=='l'||wi=='l'||tu=='l'||wo=='l')&&(countertimewhenactoron==0))||countertimewhenactoroff==0){
```

Figure 4.4.1.1. Condition for initiate the sensor readings

4.4.2 Read inputs from sensors

If the above condition satisfies then the esp 32 will read the data from the sensors

- DHT 11 sensor connects it VCC to the 3 v of esp32 , GND is connected to ground of esp32 and S is the signal pin that is connected to one of the GIO pin in esp32By using the above imported DHT.h library we will define the version(DHT 11) and then we will start reading data from it using the built in function. The code snippet shows the the implementation of DHT 11 sensor

- The turbidity sensor connects it VCC to the 3 v of esp32 , GND is connected to ground of esp32 and S is the signal pin that Is connected to one of the GPIO pin in esp32 .We will take the random of 800 values from the sensor and average them for accurate value and that is converted into NTU using the formula explained in system design.
- Water level sensor connects it VCC to the 3 v of esp32 , GND is connected to ground of esp32 and S is the signal pin that Is connected to one of the GPIO pin in esp32.We will take the random of 300 values for accuracy and average them .Based on the resistance we get as input we will map the value between 0 and 100.Then output will be in range (0-100) using Map() function.
- The Do and Ph sensor are connected its it VCC to the 3 v of esp32 , GND is connected to ground of esp32 and S is the signal pin that Is connected to one of the GPIO pin in esp32.This way the respective readings are taken from sensor and given to Esp 32..

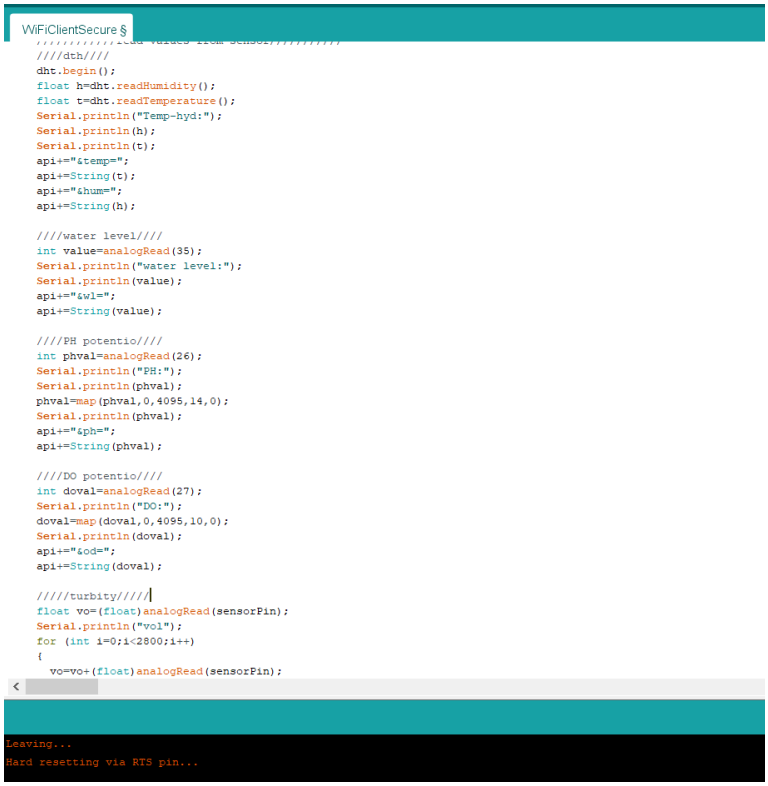

Figure 4.4.2.1. Reading data

4.5 API request to the server

An API request allows you to retrieve data from a data source, or to send data. APIs run on web servers, and expose endpoints to support the operations client applications use to provide their functionality. Each API request uses an HTTP method. The most common methods are GET , POST , PATCH , PUT , and DELETE .

4.5.1 Create API URL and initialize request

In this phase the data is concatenated to the URL and make a API URL to send the data to server from client.

We can see the code snippet below.

```
client.println(String("GET https://aquasystem.000webhostapp.com/test.php?")+String(apireq)+String(" HTTP/1.0"));
 client.println("Host: aquasystem.000webhostapp.com");
 client.println("Connection: close");
 client.println();
```

Figure 4.5.1.1. Creating API request

4.5.2 Receive Response

In this phases, the response from the server is taken and used for further analysis in esp32.The response is read character by character ,for that we use the while loop to read until the response available. After reading the response we will close the connection using client.stop() function.

```
while (client.connected()) {
  String line = client.readStringUntil('\n');
  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
}
// if there are incoming bytes available
// from the server, read them and print them:
while (client.available()) {
  char c = client.read();
  b[i]=c;
  i=i+1;
  Serial.write(c);
}
client.stop();
```

Figure 4.5.2.1. Receiving response.

4.6Storing Data into Database

4.6.1 Setting up Database in the server.

Before dumping the values into the database we need to create a data base in server and create a table for the storing of values.

So we are using Mysql as the database Structured query language. So lets know about Mysql ????

4.6.1.1 What is Mysql ?

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more. Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded application

4.6.1.2 Creating table in database

We are creating table in the database of the server using Sql query

So the queries shown below:

```
Create table data(ID int,NAME varchar(20),status int ,Pin float,val float);

INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (2,'waterin',0,21,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (3,'waterout',0,22,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (4,'turbine',0,26,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (5,'ph',5.6,30,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (6,'do',0,31,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (7,'turbidity',0,32,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (8,'temp',0,33,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (9,'humidity',0,33,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (10,'buzzer',0,27,0);
INSERT INTO `data`(`ID`, `NAME`, `status`, `pin`) VALUES (11,'waterlevel',0,25,0);
```

Figure 4.6.1.2.1. Creating table and inserting device row.

In the above table we are creating definition using DDL language create and later inserting the rows defining each as one sensor or actor to store status and values.

4.6.2 Create a webpage in server

In order to receive the requested data from the esp32 we need a server side scripting language to take data from request .So we are using PHP.

4.6.2.1 What is PHP?

PHP is a server side scripting language. That is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only.

We have created a PHP page that is “test.php”In this page we are receiving data from URL using $_GET[” ”] function and access the respective variable data.After receiving the values in order to update the Database connection is established using PHP file conn.php and it is updated to the database using the update command in Sql. After updating the values the database connection need to be closed.

```php
<?php
$mysqli = new mysqli("localhost","id16841383_aqua","NeerajSiri@34.","id16841383_aquasystem");

?>
```

Figure 4.6.2.1.1. initialize connection to database.

/public_html/test.php

```php
}

  ///// reading sensor data/////

if(isset($_GET['ph'])){
    $ph=$_GET['ph'];
    $sql="UPDATE data SET val=".$ph." where id=5 ; ";
    $result =mysqli_query($mysqli,$sql);
    echo "ph".$ph."\n";
}
else{

   // echo"no ph";
}


if(isset($_GET['tur'])){
    $tur=$_GET['tur'];
    $sql="UPDATE data SET val=".$tur." where id=7 ; ";
    $result =mysqli_query($mysqli,$sql);
    echo "tur".$tur."\n";
}else{

    //echo"no tur";
}
```

Figure 4.6.2.1.2. Receiving data from request

```php
mysqli_close($mysqli);
```

Figure 4.6.2.1.3. Closing connection of database

After updating the values we will send the response from the server to esp32. That response is contains flag values of actors in .json format which is used for control system in esp32. As shown below

```
echo "{a:".$a.",i:".$i.",o:".$o.",t:".$t."}" ;
```

Figure 4.6.2.1.4. Creating response for esp32 from server.

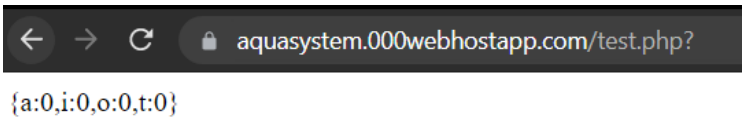

Figure 4.6.2.1.5. Response format to esp32

4.7 Showing data to user

The monitored values are shown in the website. The values from the database are updated for every 20 seconds to be accurate. We used HTML,CSS,JS for showing the data in a frontend.

4.7.1 What is HTML ?

HTML is a Markup language that are used to create electronic documents, especially pages on the World Wide Web that contain connections called hyperlinks to other pages. Every web page you see on the Internet, including this one contains HTML code that helps format and show text and images in an easy to read format. Without HTML a browser would not know how to format a page and would only display. Web browser receives HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the

building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page

4.7.2 What is CSS?

CSS is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML,Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications. 34 CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple.

4.7.3 What is JS ?

JavaScript HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behaviour and all major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven , functional and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host

environment (usually a web browser) provides those APIs.The code shown below (HTML,CSS,JS) for front end and PHP for backend.

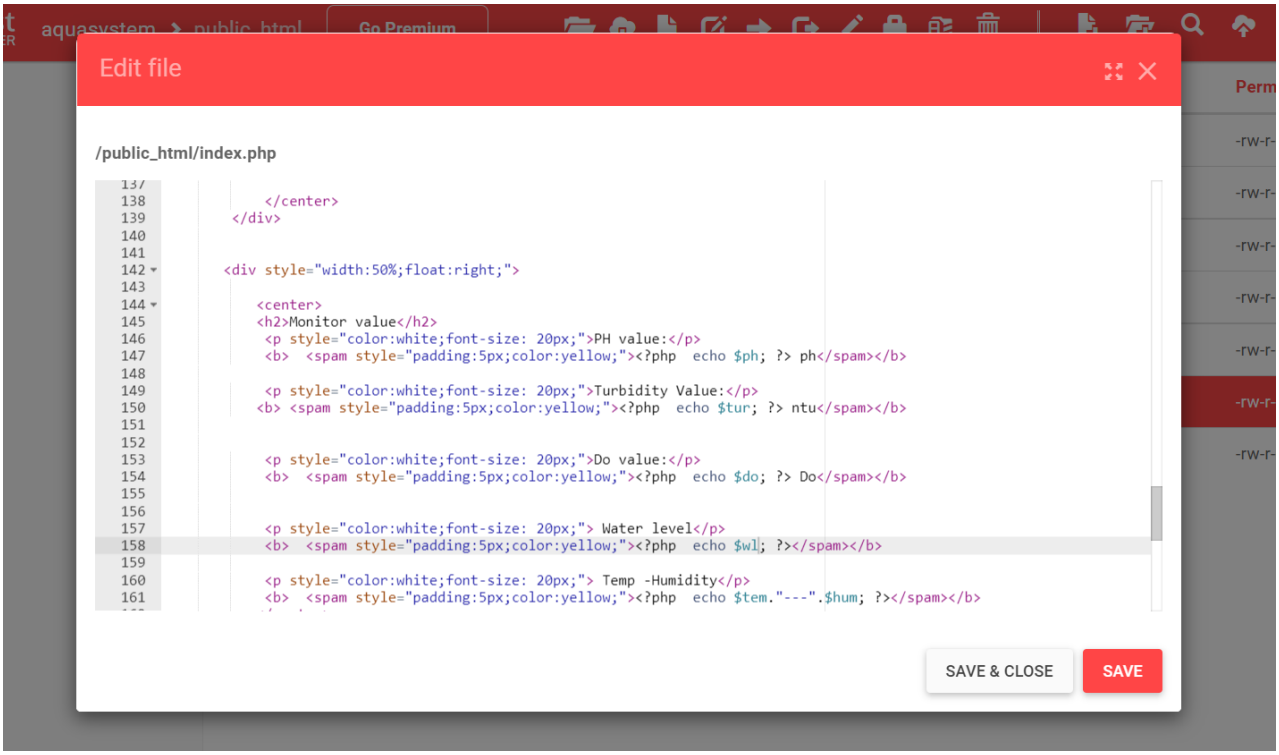

Figure 4.7.1 HTML,CSS for frontend

The website updates itself for every 20 seconds ,for new database values.we have used JS as shown below to make updates.

```
<script>
    function myFunction(x,y) {
        var k="http://aquasystem.000webhostapp.com/dataresponse.php?"+"id="+x+"&action="+y;
        location.replace(k);

    }

    setTimeout(function () {
          location.reload();
        }, 20 * 1000);
</script>
```

Figure 4.7.2. updating for every 20 senconds.

4.8 Alerting user

In this the user is alerted using the alert message from the website based on the range of tolerance. We have given condition that generates the alert in website when user visits.

```
<?php

if($tem>=42){
    echo '<script>alert("Temp is high");</script>';
}
if($tur>=900){
    echo '<script>alert("Turbidity is more");</script>';
}
if($do<=5){
    echo '<script>alert("Oxygen in low");</script>';
}
if($wl>=90){
    echo '<script>alert("Water level is high");</script>';
}
if($wl<=20){
    echo '<script>alert("Water level is Low");</script>';
}
?>
```

Figure 4.8. Setting alert tolerance in website

CHAPTER-5

OUTPUT SCREENS

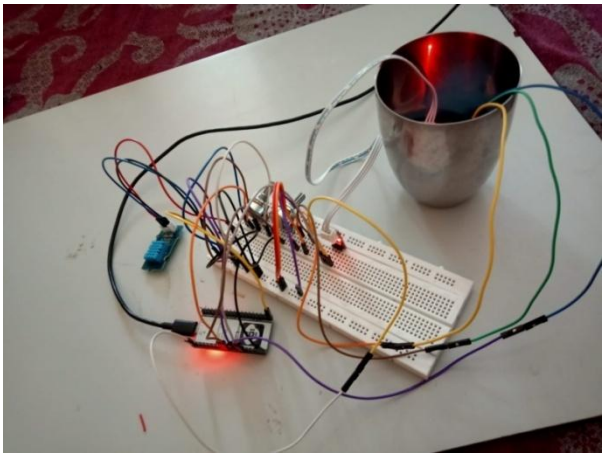

Figure 5.1 circuit testing with water sample

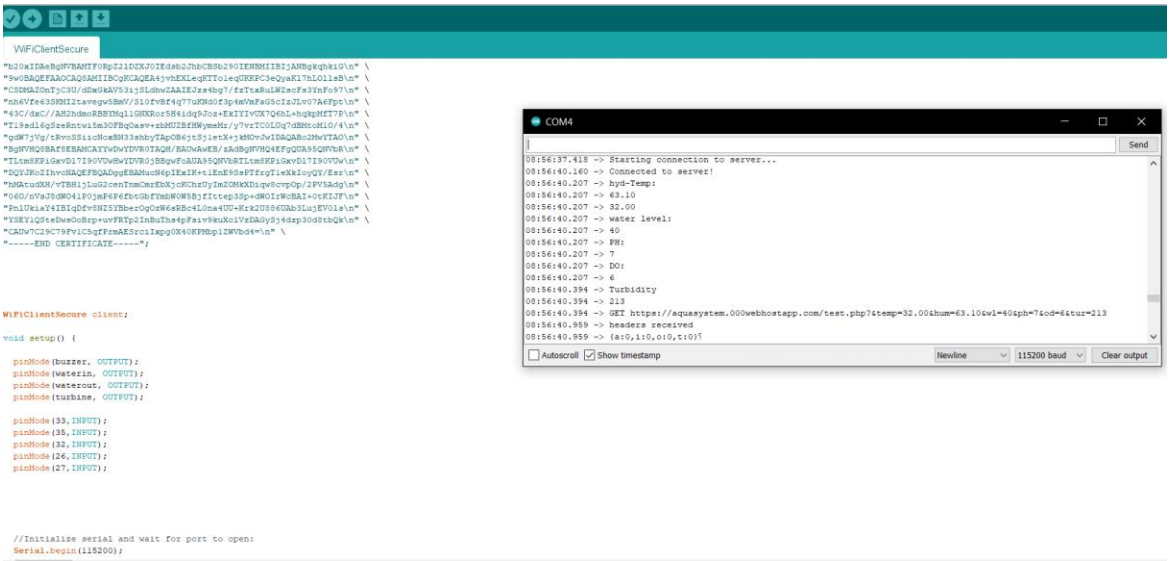

Figure 5.2 Serial monitor output.

	ID	NAME	status	pin	val
Edit Copy Delete	1	auto	0	-1	0
Edit Copy Delete	2	waterin	0	21	0
Edit Copy Delete	3	waterout	0	22	0
Edit Copy Delete	4	turbine	0	26	0
Edit Copy Delete	5	ph	1	30	7
Edit Copy Delete	6	do	1	31	6
Edit Copy Delete	7	turbidity	1	32	213
Edit Copy Delete	8	temp	1	33	32
Edit Copy Delete	9	humidity	1	33	63.1
Edit Copy Delete	10	Buzzer	0	27	0
Edit Copy Delete	11	waterlevel	1	25	40

Figure 5.3 Database values table

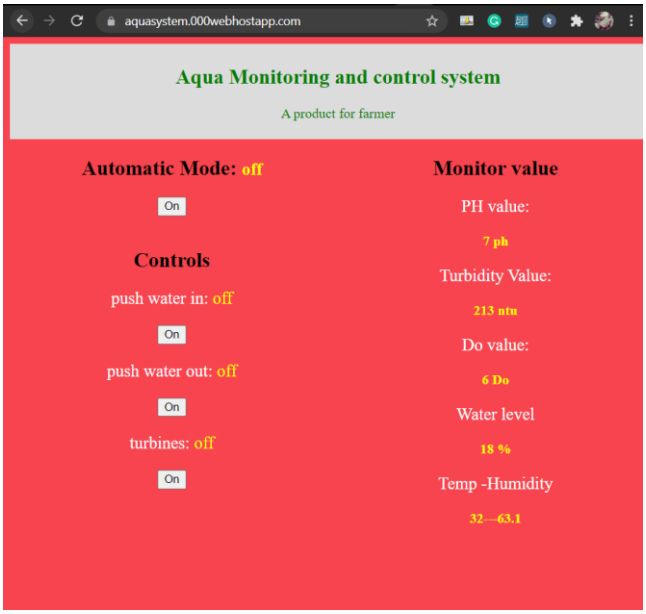

Figure 5.4 Website

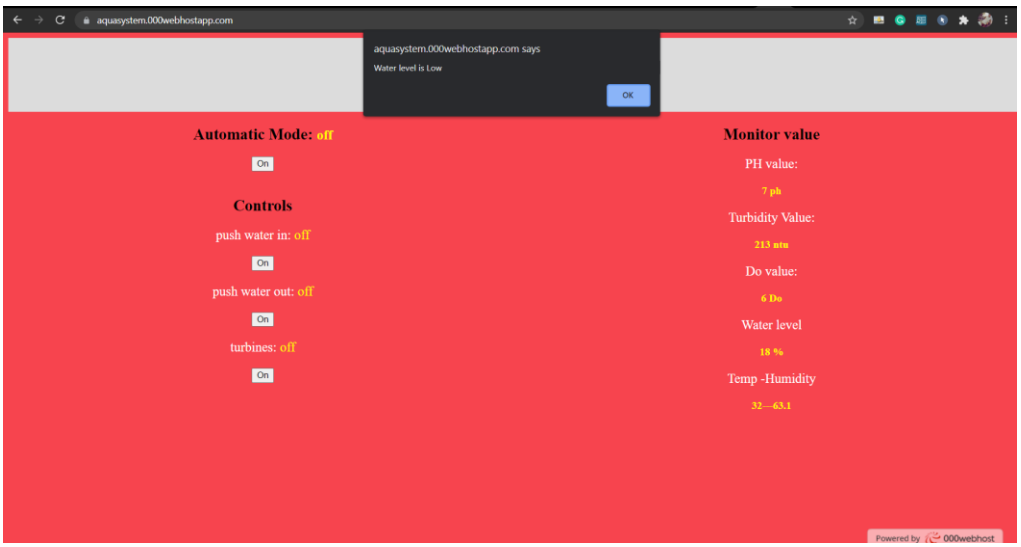

Figure 5.5 Alert Diagaram

CHAPTER-6

CONCLUSION

The "IOT BASED MONITORING SYSTEM FOR AQUA FARM" provides a user interface to the Aqua farmer which is developed using HTML, CSS, PHP and JavaScript, which fully meets the functional and non-functional requirements as specified in the software requirement specification. The ESP 32 code is done in C language and the devices are working perfectly at 5-10V Power supply.

The project is intended to replace the age old system of the Aqua farming where the quality check-up is done periodically which results in higher maintenance cost for the Aqua farmer. Using pH sensor, Temperature sensor (dht11), Turbidity sensor and Dissolved Oxygen sensors, water level sensor we can monitor the quality from website and the user can check the values and do some actions based on the readings.

FUTURE SCOPE

The future scope of this project is to improve design, implementation and documentation in such a way that anyone can use this project for better its performance. This will be extended with video surveillance, GSM module for communication and add convenience such that the devices will be vary from farmer to farmer. This project can be implemented in real time by using Arduino along with ESP32 to handle heavy devices and 220/240 V Relay as we operate with large amount power supply. This project can be extended further by providing periodic analysis of all these water parameters using graphs, pie-charts etc., for better visualization and analysis.

CHAPTER-7

BIBLIOGRAPHY

- **https://www.arduino.cc/en/guide/windows**
- **https://en.wikipedia.org/wiki/Internet_of_things**
- **https://www.w3schools.com/html/**
- **https://www.w3schools.com/Css/**
- **https://www.longdom.org/open-access/design-and-deployment-of-aqua-monitoring-system-using-wireless-sensor-networks-and-iarkick-2155-9546-5-283.pdf**
- **https://www.researchgate.net/publication/334091423_IoT_Based_Automated_Fish_Farm_Aquaculture_Monitoring_System**