INTERACTIVE AQUARIUM

Aman Khaware (1847209)
Vikash Singh (1847263)
Kumar Navin Barnwal (1847267)

Under the guidance of
Dr. J Sandeep

IOT project report submitted in partial fulfillment of the requirements of IV semester Master of Computer Applications, CHRIST (Deemed to be University)

March - 2020

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled **Interactive Aquarium** *is a bona fide record of work done by* ***Aman Khaware (1847209), Vikash Singh (1847263) and Kumar Navin Barnwal (1847267)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of IV Semester Master of Computer Applications during the year 2020.*

Head of the Department **Project Guide**

Valued-by:

	Name	:	
1.	Register Number	:	
	Examination Centre	:	CHRIST (Deemed to be University)
2.	Date of Exam	:	

ACKNOWLEDGEMENTS

First of all, we thank the lord almighty for his immense grace and blessings showered on me at every stage of this work. we are greatly indebted to our Head of the Department of Computer Science, Prof. Joy Paulose, CHRIST (Deemed to be University) for providing the opportunity to take up this project as part of my curriculum.

We are greatly indebted to our MCA Course Coordinator, Prof. Tulasi, for providing the opportunity to take up this project and for helping with her valuable suggestions.

we are deeply indebted to our project guide Dr. J Sandeep for his assistance and valuable suggestions as a guide. He made this project a reality.

we express our sincere thanks to Dr. Arul Kumar N, and Dr. Gobi R, lecturers of the Department of Computer Science, CHRIST (Deemed to be University), for their valuable suggestions during the course of this project. Their critical suggestions helped us to improve the project work.

Acknowledging the efforts of everyone, their chivalrous help in the course of the project preparation and their willingness to corroborate with the work, their magnanimity through lucid technical details lead to the successful completion of my project.

we would like to express our sincere thanks to all our friends, colleagues, parents and all those who have directly or indirectly assisted during this work.

ABSTRACT

The basic job of any aquarium owner is to feed the fishes, clean the water based on the quality of water, temperate and frequently process the required medication. Its maintenance is one of the crucial tasks. The purpose of this project is to build an Interactive Aquarium with the target customers as hotel, restaurant, office employee or any individual owning a Fresh Water Aquarium and wants to automate the task of maintenance. The system is supposed to monitor the physical changes in the water. The product will provide reading of temperature control, pH level etc. both in textual and audio form (for blind people).

The purpose of the project is to monitor the water quality of an aquarium based on PH level, chlorine, alkaline, temperature, etc. Tracking the observatory change obtained from the sensors and informing the user for any unusual activity in the form of text through LCD screen and audio. The project is to be implemented with the purpose of reducing the maintenance cost of an aquarium given that the Aquarium interacts with the user. Even blind user can experience the status of an aquarium through audio.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 OVERVIEW OF THE SYSTEM

An aquarium is a tank wherein all water creatures such as fishes, rocks that fishes need for shelter, plants that they use as food and many more. It is used for different marine purposes and others use it as decor in their home. Different types of aquariums are used for decor such as saltwater aquarium, freshwater aquarium, etc. The maintenance of saltwater is very high and so as freshwater aquarium. This project is useful for the one who wants to have a freshwater aquarium in their home but cannot afford it due to high maintenance costs. Most of the customers are unaware of the steps to be taken for maintaining a healthy aquarium for the fishes.

An interactive aquarium is designed for the customers to cut down the maintenance cost to half and maintain the aquarium themself. An aquarium gets contaminated due to multiple factors such as fishes releasing waste, water gets oily due to food provided by the customer to the fish, blooding due to fighting with other fishes disturbs the quality of water.

To check the quality of water required for fish is monitored using a pH sensor. When the quality of water is not healthy for the fishes' user gets the notification. Another most important factor for the fishes is the humidity and temperature inside an aquarium. Whenever temperature raises or decreases, notification is sent to the user's app. For feeding fish, a motor is installed with this project which will provide food to fish every 8 hours.

1.2 BLOCK DIAGRAM

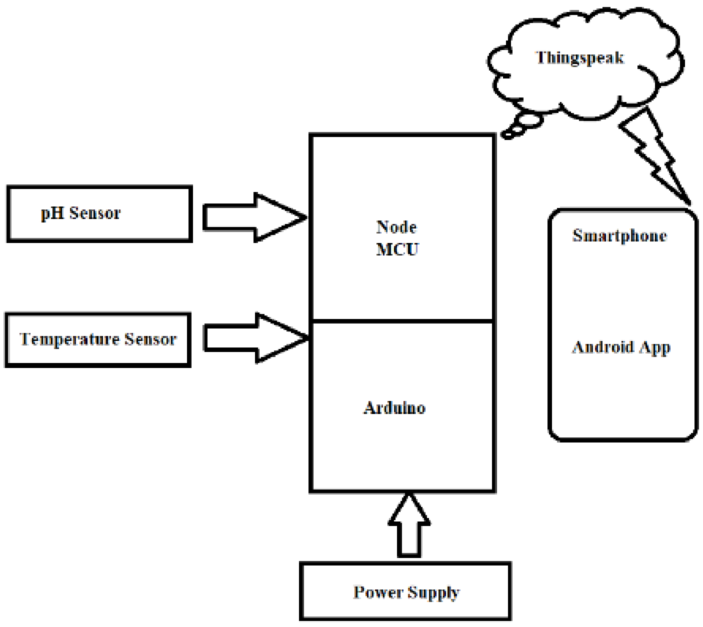


Figure 1 Block Diagram

2. SYSTEM REQUIREMENTS

The project “Interactive Aquarium” system helps the users to identify the measures need to be taken when water gets contaminated for the fishes. Adjust the temperature and humidity of the aquarium suitable for the fishes. Automatically feed the fishes every 8 hours. This application can give a quick report which will help user in assessing how frequently water was changed.

2.1 System Interfaces

The system consists and interacts with the following hardware components and software. The system is made of the following hardware components and the following software platforms are used to create the project.

2.2 Hardware Interfaces

The project uses the following hardware devices. Some of the requirements of the system are-

- Arduino Uno Board
- pH Sensor Board
- pH sensor electrode
- LCD Monitor
- Servo Motor
- Node MCU
- DHT11 sensor

2.2.1 Arduino Uno Board-

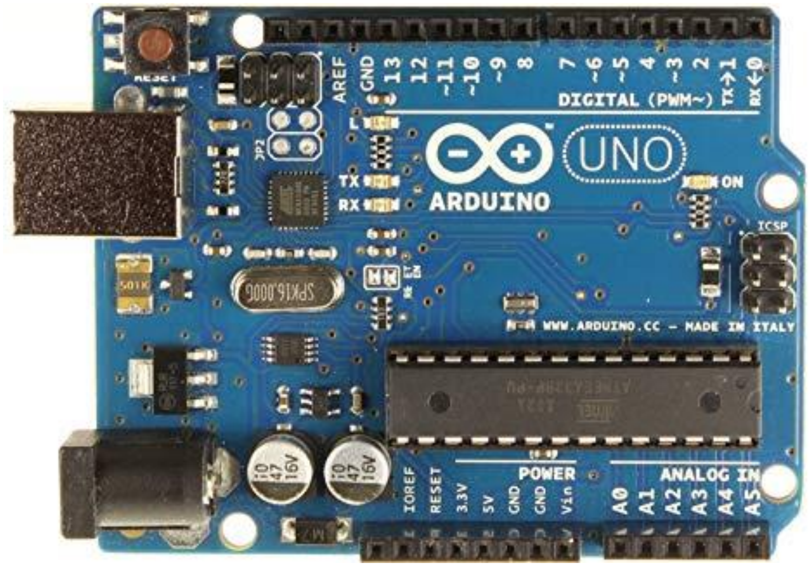


Fig. 2.1 Arduino Uno

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.[4] It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

General pin functions of Arduino Uno are-

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN**: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

- 3V3: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND: Ground pins.
- IOREF: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- Reset: Typically used to add a reset button to shields which block the one on the board.

Special pin functions of Arduino Uno-

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions2. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, label led A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.[7]

In addition, some pins have specialized functions:

- **Serial / UART**: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM** (**P**ulse **W**idth **M**odulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function**.**
- **SPI** (**S**erial **P**eripHeral **I**nterface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI** (**T**wo **W**ire **I**nterface) / I²C: A4 or SDA pin and A5 or SCL pin.

- AREF (Analog REFerence): Reference voltage for the analog inputs.

2.2.2 pH Sensor Board-

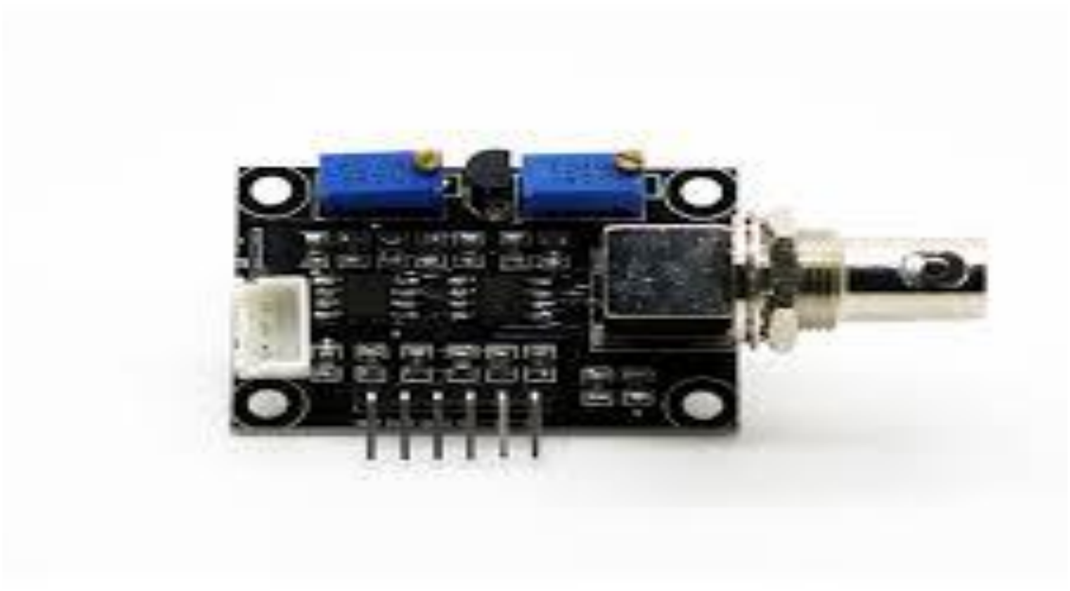


Fig. 2 2 pH Sensor Board

The analog pH sensor, specially designed for Arduino controllers is easy to use and can be used as a plug and play solution to measure pH value of a solution without any additional circuit required.

It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface.

To use it, we just connect the pH sensor with BNC connector, and plug the PH2.0 interface into the analog input port of any Arduino controller. With a simple program to ready analog voltage, and we will get the pH value easily.

<u>Working of pH Sensor</u>

The pH is a measure of the acidity or alkalinity of a solution; the pH scale varies from 0 to 14.The pH indicates the concentration of hydrogen ions [H]+ present in certain solutions.

It can be quantified accurately using a sensor that measures the difference of potential between two electrodes: a reference electrode (silver/silver chloride) and a glass electrode is sensitive to hydrogen ion. This is what will form the probe. In addition, there are that use an electronic circuit to condition the signal appropriately, and that we can use this sensor with a microcontroller.

2.2.3 pH Sensor Electrode-

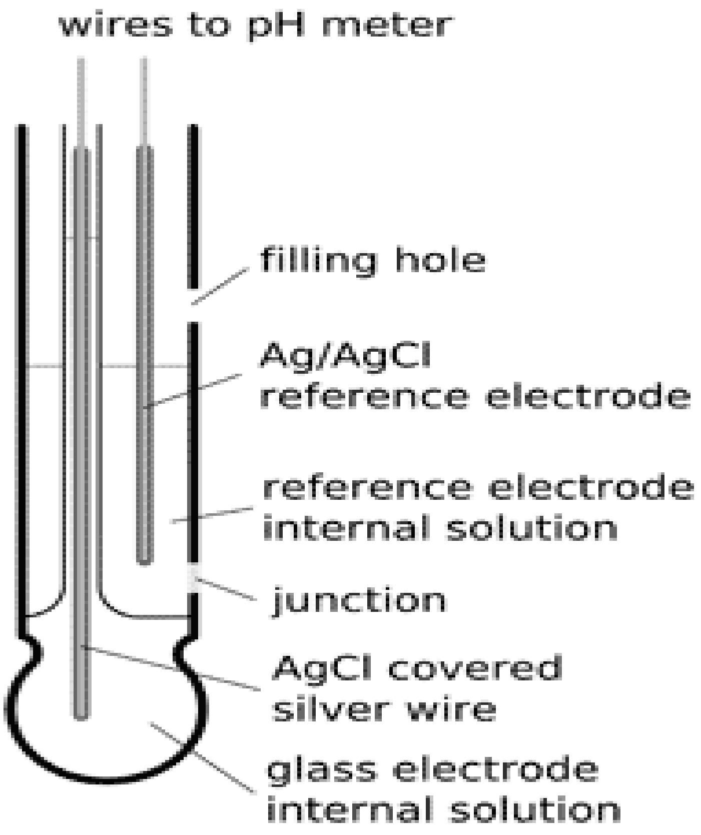

wires to pH meter
filling hole
Ag/AgCl reference electrode
reference electrode internal solution
junction
AgCl covered silver wire
glass electrode internal solution

Fig. 2 3 pH Sensor Electrode

pH electrodes are constructed from a special composition glass which senses the hydrogen ion concentration. This glass is typically composed of alkali metal ions that undergo an ion exchange reaction with the hydrogen ions in the test solution to generate a potential difference. The bulb is filled with an acid solution (e.g. 0.1 $molL^{-1}$ HCl).

2.2.4 Humidity and Temperature sensor (DHT 11)-

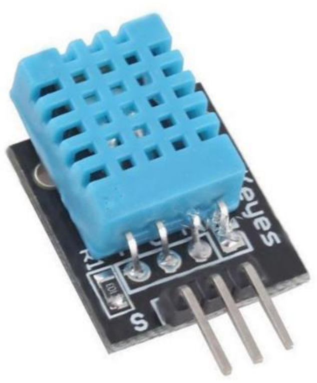


Fig. 2 4 DHT11 Sensor

The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%.

2.2.5 Servo Motor

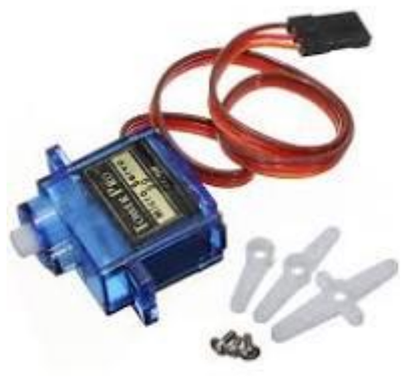


Fig. 2 5 Servo Motor

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages.

2.2.6 NodeMCU

Fig. 2.6 NodeMCU

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.

2.2.7 Jumper Wires-

Fig. 2 7 Jumper Wires

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

- Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of

  both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
- Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
- Banana connectors – are commonly used on test equipment for DC

  and low-frequency AC signals.
- Registered jack (RJnn) – are commonly used in telepHone (RJ11) and computer networking (RJ45).
- RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.
- RF connectors – are used to carry radio frequency signals between circuits, test equipment, and antennas.

2.2.8 LCD Display

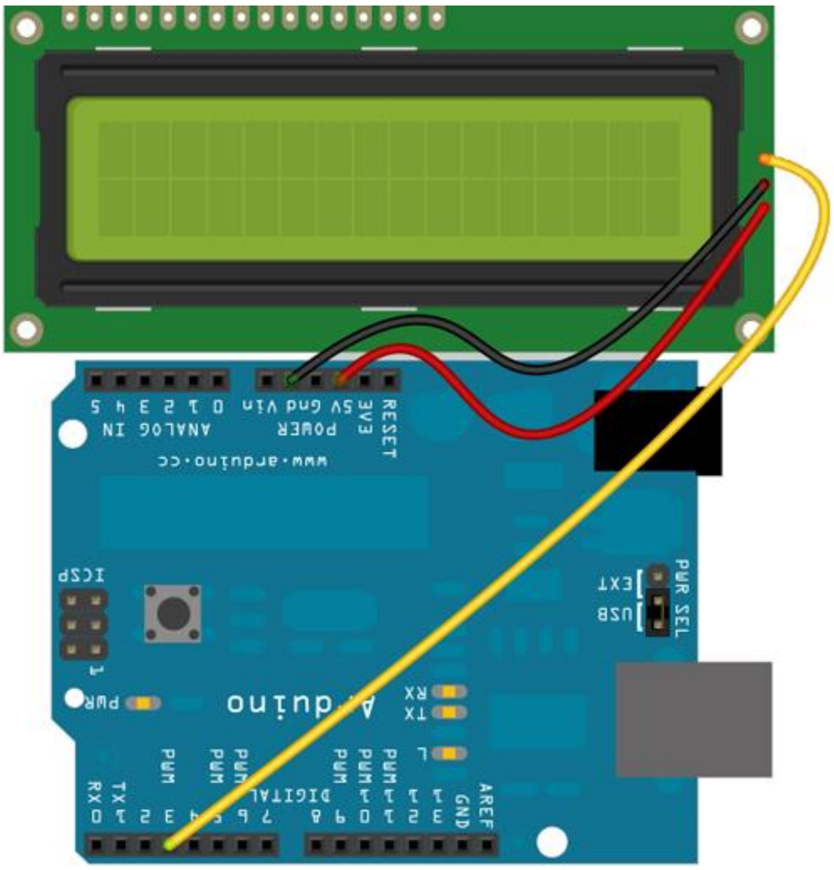


Fig. 2.8 LCD Display

LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation. LEDs have a large and varying set of use cases for consumers and businesses, as they can be commonly found in smartpHones, televisions, computer monitors and instrument panels. LCDs were a big leap in terms of the technology they replaced, which include light-emitting diode (LED) and gas-plasma displays. 16×2 LCD is named so because; it has 16 Columns and 2 Rows.

2.3 Software Interfaces

The System requires some of the software interfaces and support, this includes the following software services and applications.

2.3.1 Arduino IDE

The Arduino IDE allows you to write code and upload sketches to any official Arduino board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on processing and other open-source software. The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

2.3.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development. the following features are provided in the current stable version: Gradle-based build support, Android-specific refactoring and quick fixes, Lint tools to catch performance, usability, version compatibility and other problems, ProGuard integration and app-signing capabilities, Template-based wizards to create common Android designs and components, A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations, Support for building Android Wear apps, Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine and Android Virtual Device (Emulator) to run and debug apps in the Android studio.

2.3.3 HTML

HTML stands for Hyper Text Markup Language and was intended to be used solely to provide document structure for a text-based communication medium. Hypertext Markup

Language commonly referred to as HTML is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create

structured documents by denoting structural semantics for text such as headings, paragrapHs, lists, links, quotes and other items.

2.3.4 BOOTSTRAP

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typograpHy, forms, buttons, navigation and other interface components. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps.

Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as source code. Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto- complete function for input fields.

2.3.5 JAVASCRIPT

JavaScript can function as both a procedural and an object-oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects. JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via eval introspection (via for ... in), and source code recovery (JavaScript programs can decompile function bodies back into their source text). JavaScript is also used in environments that are not Web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client-side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation. It is also used in game development, the creation of desktop and mobile applications, and server-side network programming with run-time environments such as Node.js.

2.4 Cloud Software Interfaces

The following cloud services and database interfaces are being used to develop this application. The main cloud device that interfaces with the IOT device is the Think Speak IOT cloud. The google firebase is also used to store the details for the application.

2.4.1 Think speak IOT cloud

ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates". ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications. ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks. ThingSpeak has a close relationship with Mathworks, Inc. All of the ThingSpeak documentation is incorporated into the Mathworks' Matlab documentation site and even enabling registered Mathworks user accounts as valid login credentials on the ThingSpeak website.

2.4.2 Google Firebase

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment. Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud.

A mobile application will integrate the application and the IOT components and the cloud where data will be stored. The data from the sensors stored in the cloud storage will be used by the application to give the necessary information to the users. Additional support can be given to users like the analysis of the data that is stored. This includes frequency at which water has been changed, usage statistics and so on.

2.5 Product Functions and Characteristics

The main function of this Interactive Aquarium is that it keeps track of water quality and humidity & temperature required for fresh water aquarium. When the temperature & humidity or water pH value is low/higher than the certain threshold value of pH and temperature, an application will give an alert to the user.

A mobile application will integrate the application and the IOT components and the cloud. The data from the sensors stored in the cloud storage will be used by the application to give the necessary warnings and notifications to the users. Additional support can be given to users like the analysis of the data that is stored. This includes the water changed frequency, frequently temperature and humidity was adjusted.

2.6Constraints

The constraints this system will face include multiple device required for this project. This project requires active data connection, to give the notification in the mobile device to the user. Bluetooth connectivity is required to give audio notification to the user at home.

2.7 Design Constraints

The design constraint include the accuracy of the data received from pH sensor and humidity & temperature sensor(DHT11). The temperature outside and inside the aquarium will vary. Sensing accurate temperature and convert the wave into digital signal will be a challenge as the sensor will be placed inside an aquarium. If there is any leakage it will give a user incorrect data.

2.8Modular Specification

Module 1 – Determining the pH value of the water

This module determines the pH value of the water by using the pH sensor electrode. pH is a measure of hydrogen ion concentration, a measure of the acidity or alkalinity of a solution. The pH scale usually ranges from 0 to 14.

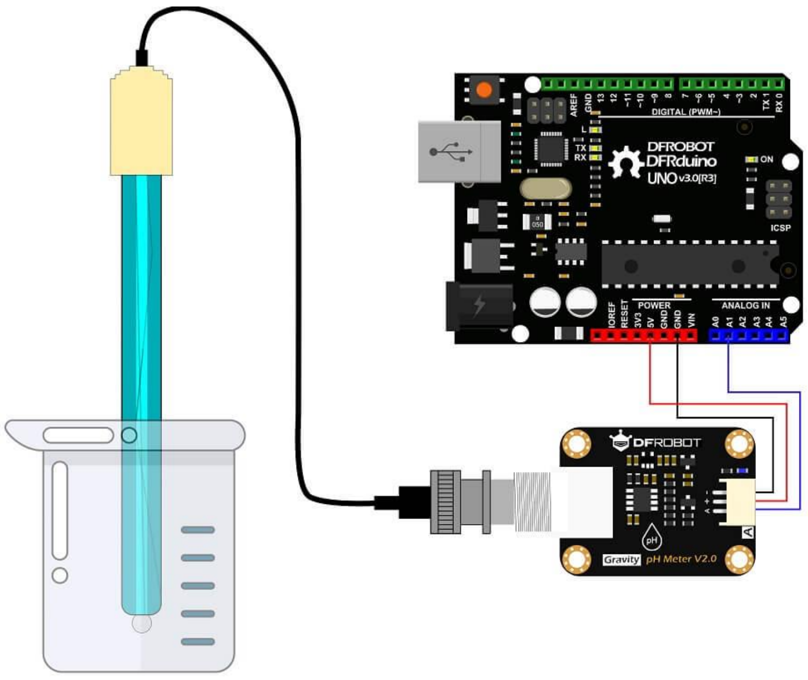


Fig. 2.9 pH sensor

Fig. 2.9 sense data from water and send the information to the cloud using NodeMCU.

Module 2 – Determining the humidity and temperature of an aquarium

This module determines the humidity and temperature inside an aquarium for the fishes. When water temperature is high then warm water causes fish to become more active and

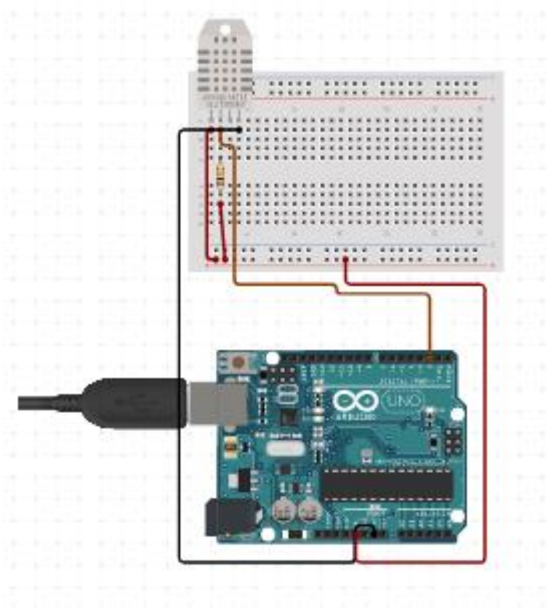


Fig. 2.10 DHT11 Sensor

require more oxygen. Warm water actually holds *less* oxygen than cooler water. In severe cases, there won’t be enough oxygen to go around, and the fish can suffocate. Fig. 2.10 sense data of humidity and temperature and the result in cloud.

Module 3 – Feeder for feeding fish after every 8 hours

This module provides food to the fishes after a set amount of time. In our case every 8 hours’ feeder will move and it will drop the food for the fishes.

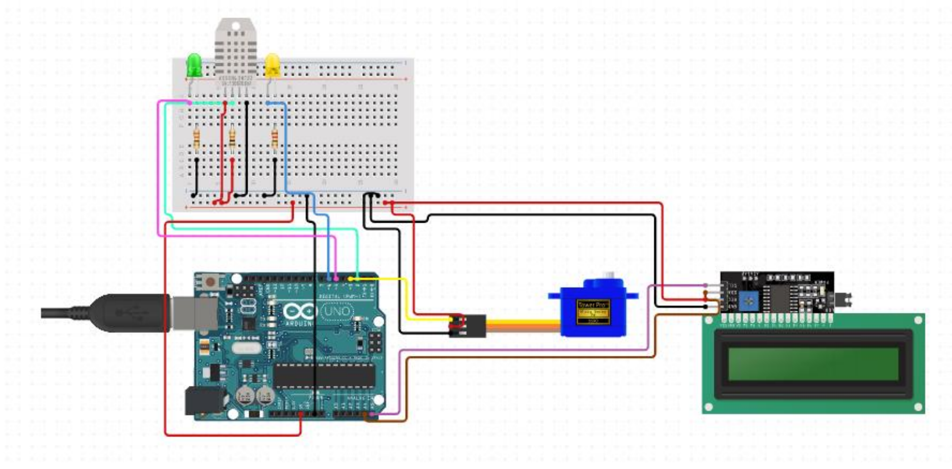


Fig. 2.11 Servo Motor

Fig. 2.11 is a feeder that will provide food to fish every 8 hours.

2.9 SCHEDULE AND ESTIMATION

The project was started in the month of November and planning to be done by end of February.

Schedule for the project: -

Week 1: Synopsis details.

Hardware details.

Circuit diagram.

Week 2: Detailed problem analysis and a literature survey.

Hardware details.

Hardware tools specification.

Week 3: The working of design.

Components requirement study.

Discuss the Connection.

Week 4: User interface code.

Week 6: Serial input/output

Paperwork.

Report.

Week 7: Testing of user interface code.

Design report.

Week 8: Validation and testing.

Draft report.

Week 9: Enhancement of design and serial interface.

Complete working project.

3. DESIGN SPECIFICATION

3.1 User Interface

It is always challenging for someone to operate any new tool or technology. For the new as well as any normal users' it become very important that they should be able to operate and understand the new tools/technology provided to them. Data fetched from the sensors and sent to the cloud is need to be presented in a presentable format so that every user can understand the analysis generated from the data collected from the sensors and also to make it easy for the user the understand and operate system.

The purpose of our system is to provide information of aquarium condition 24/7 hours to the user. Many users love to have an aquarium at their home, hotels etc. But all cannot afford to have an aquarium to its high maintenance cost. Our system will provide a luxury to the user to understand about their aquarium by constantly providing them information about water quality, temperature and humidity required for the freshwater fishes. Our system will provide all this information on the application that will be provided to the user. The two system has been developed. One for the admin and the other for the user. Admin's work will be to register user info along with the aquarium ID which will be auto generated. And the admin can view all the complaints generated by the user's end.

Web Application Interface: -

1. Login Page

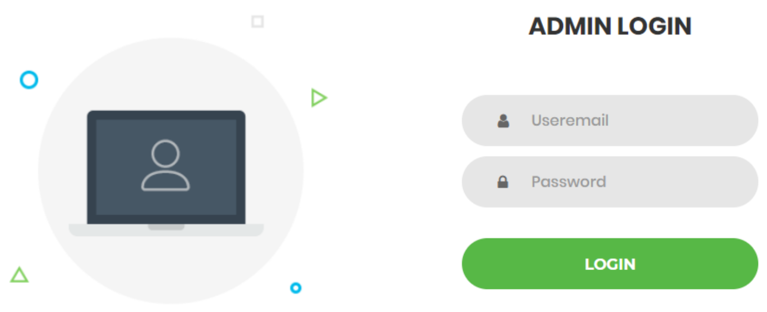


Figure 2.1 Login Page

Fig 3.1 is a welcome web application interface. It is login page for the admin only. Admin need to provide their ID i.e., his mail and password.

2. Dashboard

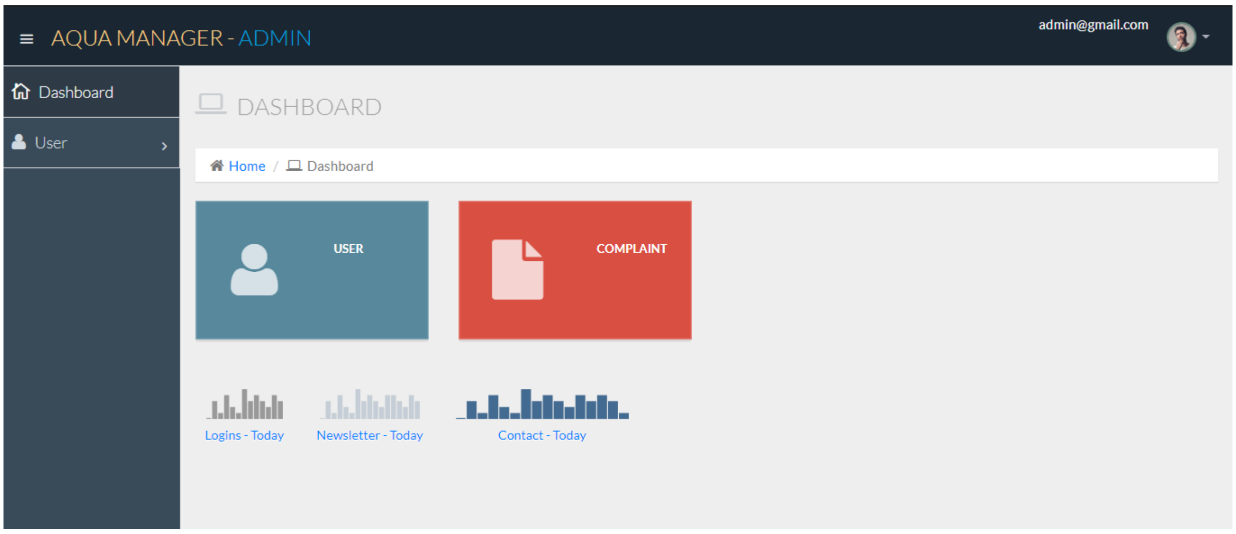


Figure3.2 Dashboard

Fig. 3.2 is a dashboard page of an admin from which user's registration will be done and they will be given unique ID.

3. Sidebar

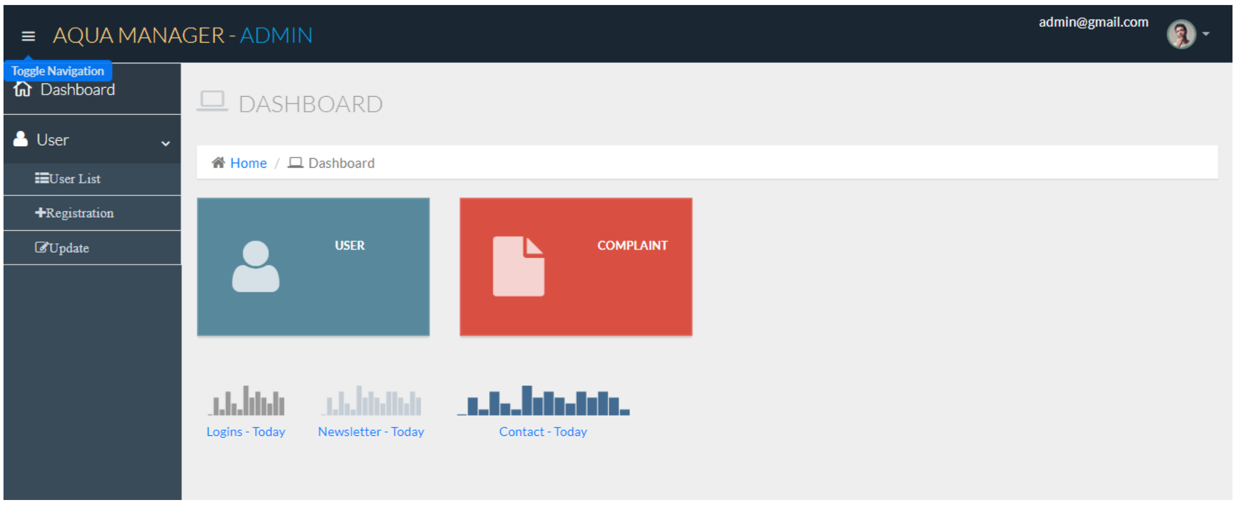


Figure 3.3 Sidebar

Fig. 3.3 will have an option of registration new user, updating old user and view the list of registered user

4. User Registration page

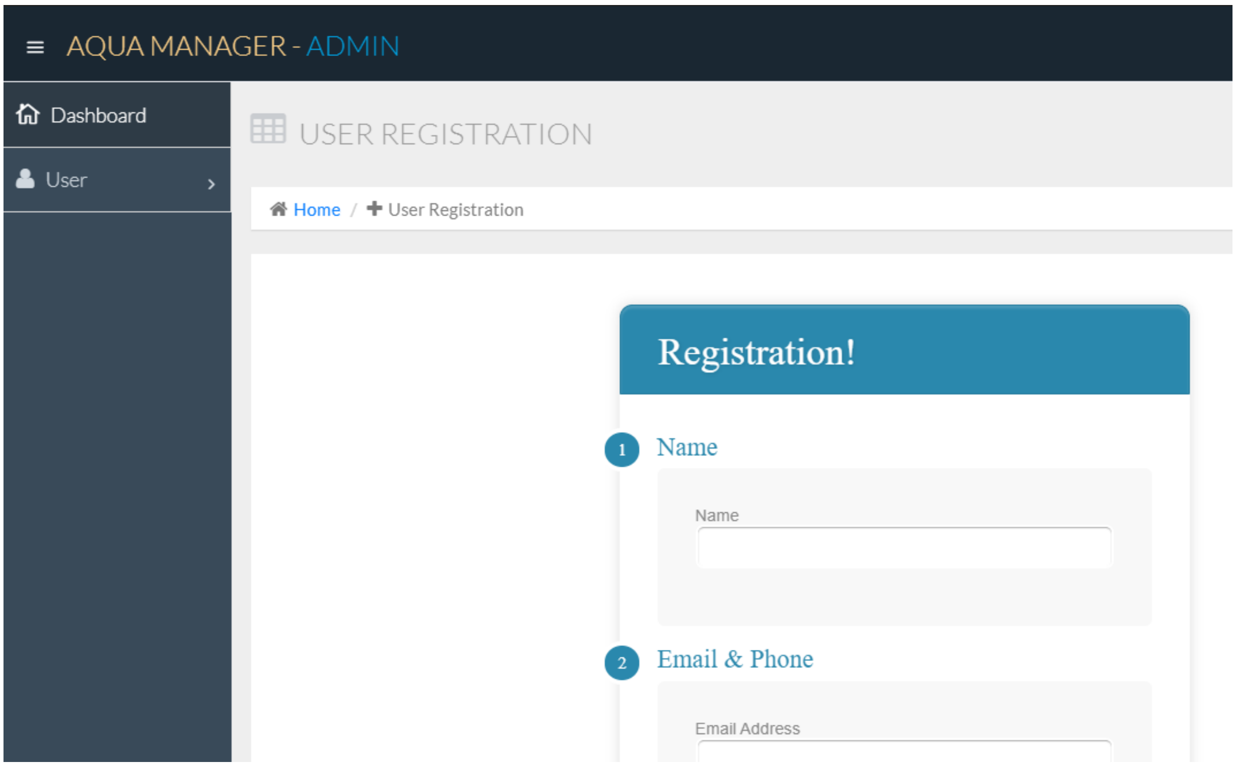


Figure 3.4 User Registration page

Fig. 3.4 is a page which will handle user registration.

5. User registered list

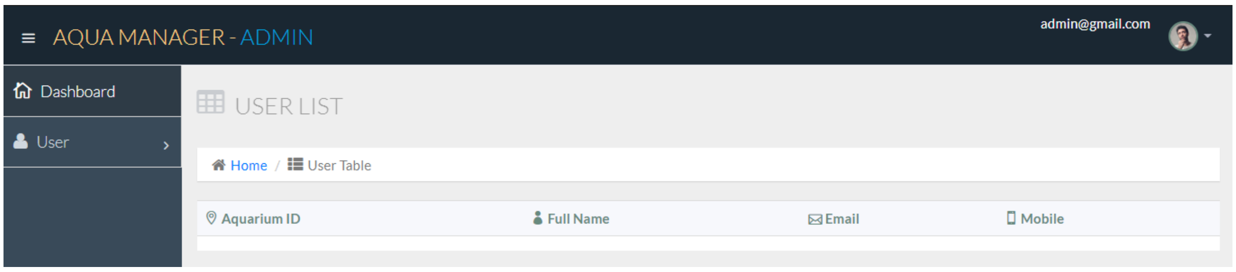


Figure 3.5 Registered user list

Fig 3.5 It will display user registered list.

Mobile Application Interface: -

1. Splash

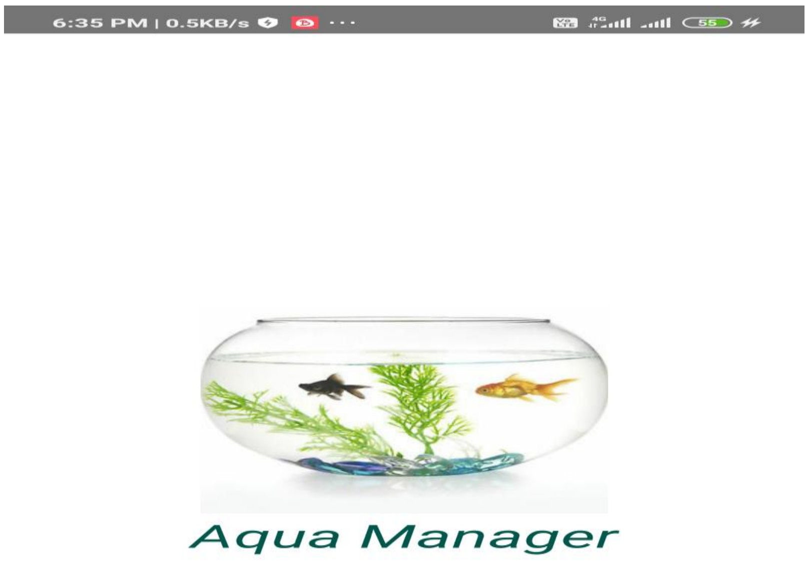


Figure 3.6 Splash Screen

Fig. 3.6 shows the welcome screen of the application. The screen contains the logo and name of the application

2. Login Screen

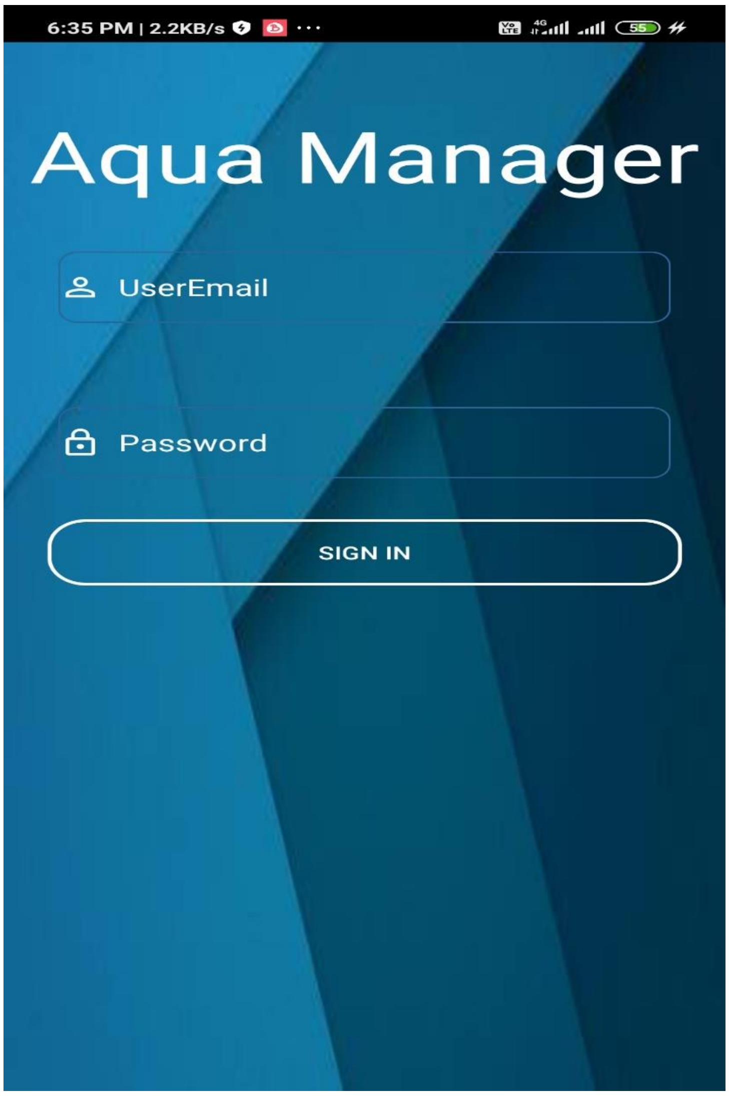


Figure 3.7 Login Page

Fig 3.7 shows the Login Screen; it acts as a login page for home-screen of the application. This screen will have a logo and a field to enter the username and password. Here users can login to the application, access features and keep track of aquarium.

3. Dashboard

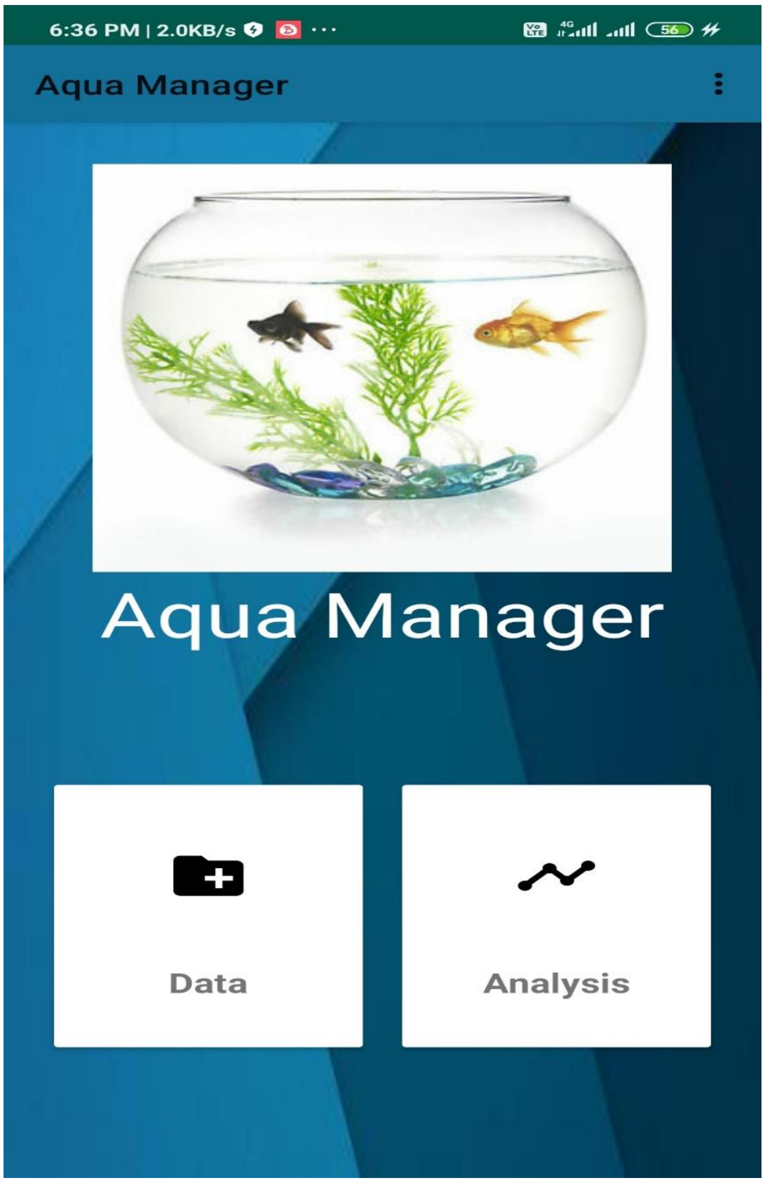


Figure 3.8 Dashboard

Fig 3.8 On click of Data, user can read the data stored in the cloud i.e., Firebase real time database and Thingspeak. On click of analysis data will be displayed in the form of grapH(Time-series).

4. App Bar menu

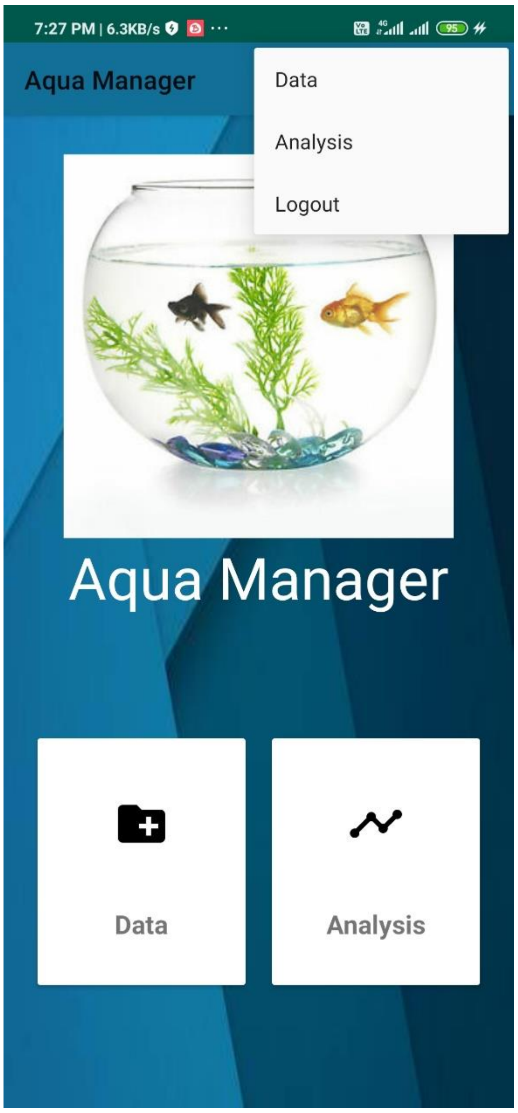


Figure 3.9 App Bar Menu

Fig 3.9 App bar menu will display certain menu for the user. On click ou logout option user will be taken to the login page.

3.2 Database Design

The following are database-design of the system. The necessary tables to store the data are included in this section. Fields required the datatypes of the same are mentioned below.

3.2.1 pH_detail

Time	**Date**	**value**
Integer	**Integer**	**Integer**

Table 3.2.1 will store the data of pH value sensed by the sensor and it will store in the table. It will take three details in the form of system date, time and value sensed by the sensor. The data will be stored in the table every 20 seconds.

3.2.2 humidity&temperature_detail

Time	Date	value
Integer	Integer	Integer

Table 3.2.2 will store the data of pH value sensed by the sensor and it will store in the table. It will take three details in the form of system date, time and value sensed by the sensor. The data will be stored in the table every 20 seconds.

3.3 Circuit diagram

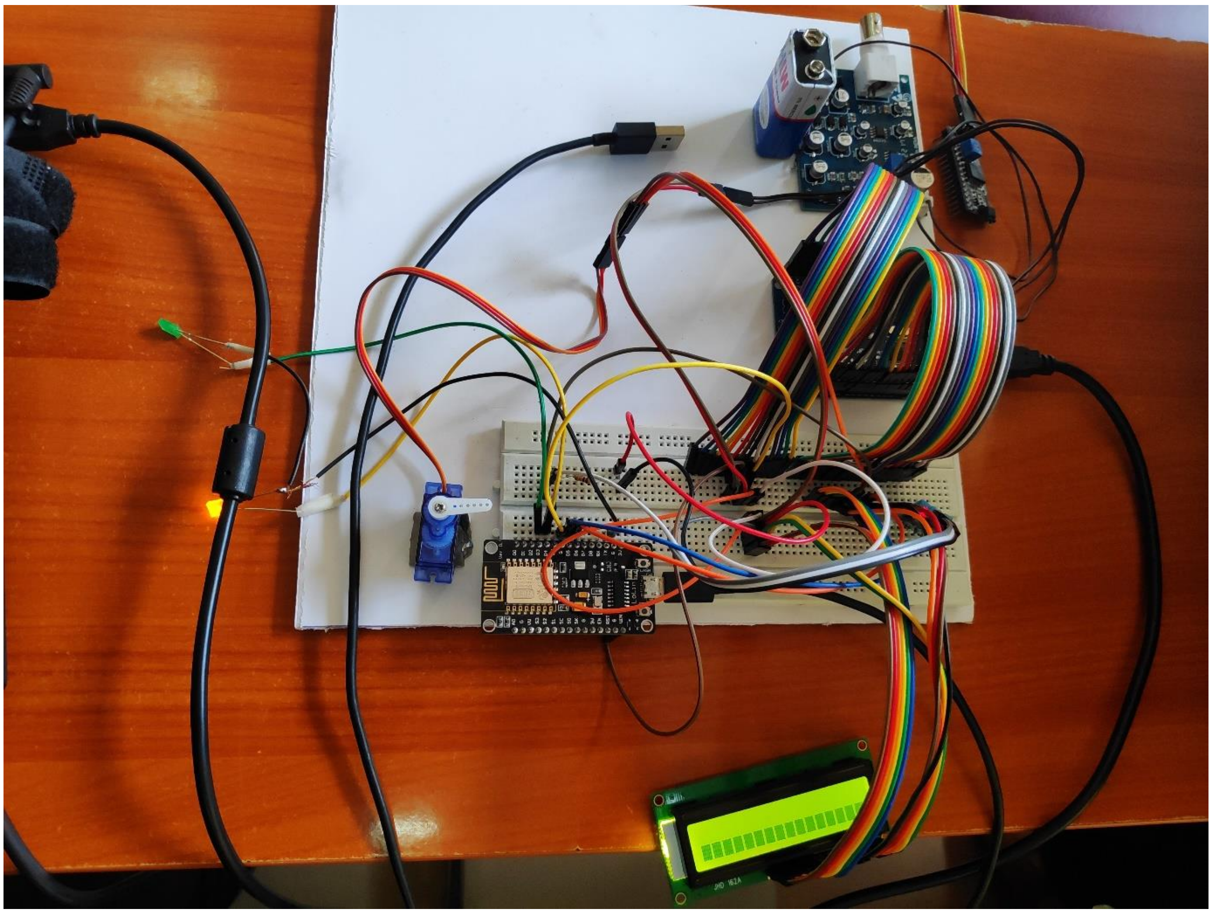


***Figure 3.10** Final Model*

Fig 3.10 is the final circuit design of the system. Arduino will be connected with the power supply and jumper wires will connect breadboard and Arduino A breadboard is a construction base for prototyping of electronics. Initially the system will check whether Wi-Fi connection is available or not. Once connection is established, pH sensor electrode measures the pH value of the water of the aquarium. The pH probe will be dipped inside aquarium and it will read pH value of water. The pH value of the water is measured and the results are will be displayed on the LCD display. When the pH level is high monitor will display the pH value of water is high. If the pH value is high the it prints message high and if the pH level is low, then it will print low pH value. The **DHT11** sensor will monitor humidity and temperature inside an aquarium for the fishes. It will print the message as high or low depending on the threshold value set. Feeder will be used for feeding fishes. Every 8-hour feeder will move and drop food for the fishes.

4. IMPLEMENTATION

4.1 TOOL/SOFTWARE DESCRIPTION

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

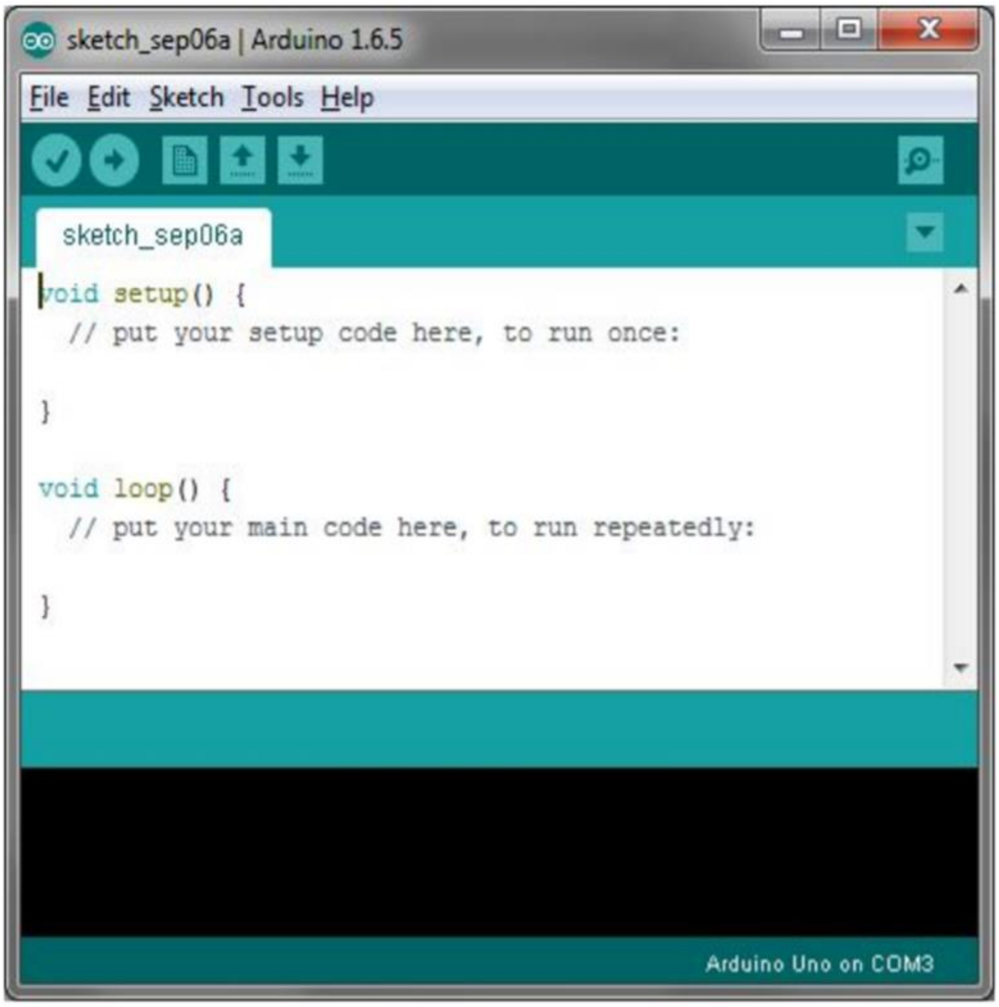
IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processingand Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for
common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2

The Arduino IDE supports the languages C and C++ using special rules of code structuring.

The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

1.Arduino IDE: Initial Setup



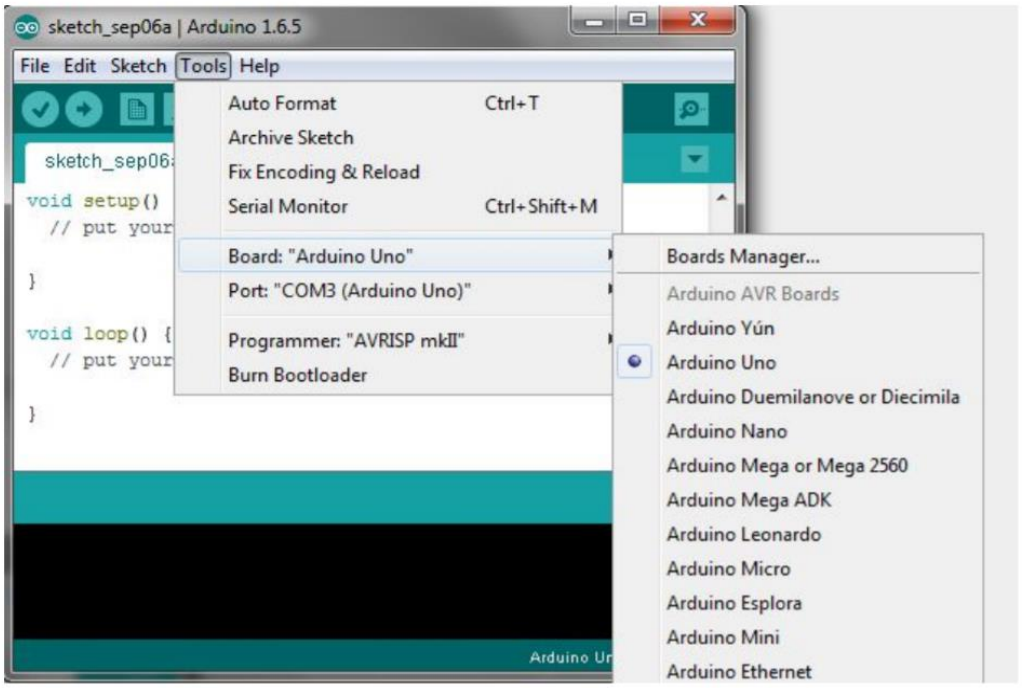
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 4.1 Arduino IDE Default Window

Fig 4.1 Download Arduino Integrated Design Environment (IDE) here (Most recent version: 1.6.5): https://www.arduino.cc/en/Main/Software. This is the Arduino IDE once

it's been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.

2. IDE: Board Setup



```
sketch_sep06a
void setup()
  // put your
}

void loop() {
  // put your
}
```

Figure 4.2 Board Setup

Fig 4.2 we have to tell the Arduino IDE what board we are uploading to. Select the Toolspulldown menu and go to Board.This list is populated by default with the currently available Arduino Boards that are developed by Arduino. If you are using an Uno or an Uno-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If we are using another board/clone, select that board.

3. IDE: COM Port Setup

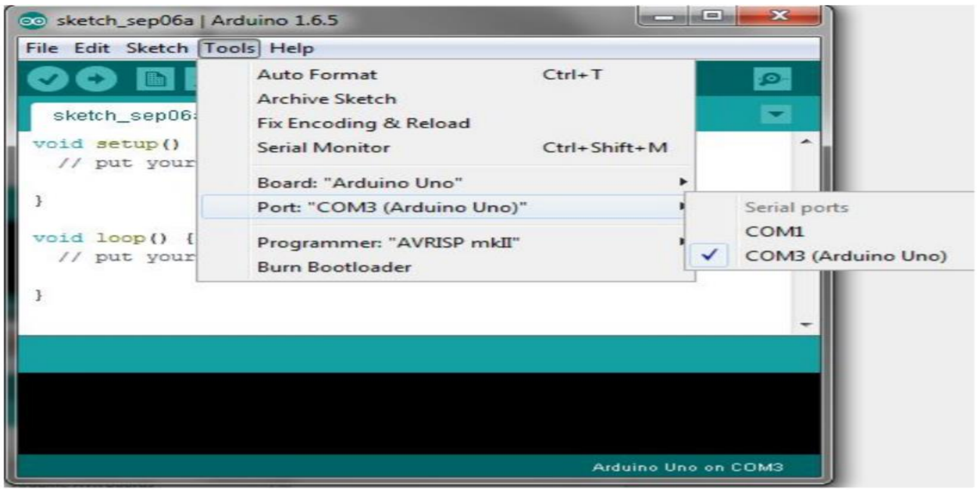


Figure 4.3 IDE: COM Port Setup

Fig 4.3 if we have downloaded the Arduino IDE before plugging in our Arduino board, when we plugged in the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Tools pulldown menu and then Port. Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give its name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, we should see the board type and COM number of the board you plan to program. Note: The Arduino Uno occupies the next available COM port; it will not always be COM3.At this point, your board should be set up for programming, and you can begin writing and uploading code.

4. Testing Your Settings: Uploading Blink

Upload Button:

Figure 4.4 Power Button

Fig 4.4 One common procedure to test whether the board you are using is properly set up is to upload the “Blink” sketch. This sketch is included with all Arduino IDE releases and can be accessed by the Filepull-down menu and going to Examples, 01. Basics, and then select Blink. Standard Arduino Boards include a surface-mounted LED labeled “L”

or "LED" next to the "RX" and "TX" LEDs, that is connected to digital pin 13. This sketch will blink the LED at a regular interval, and is an easy way to confirm if your board is set up properly and you were successful in uploading code. Open the "Blink" sketch and press the "Upload" button in the upper-left corner to upload "Blink" to the board.

\

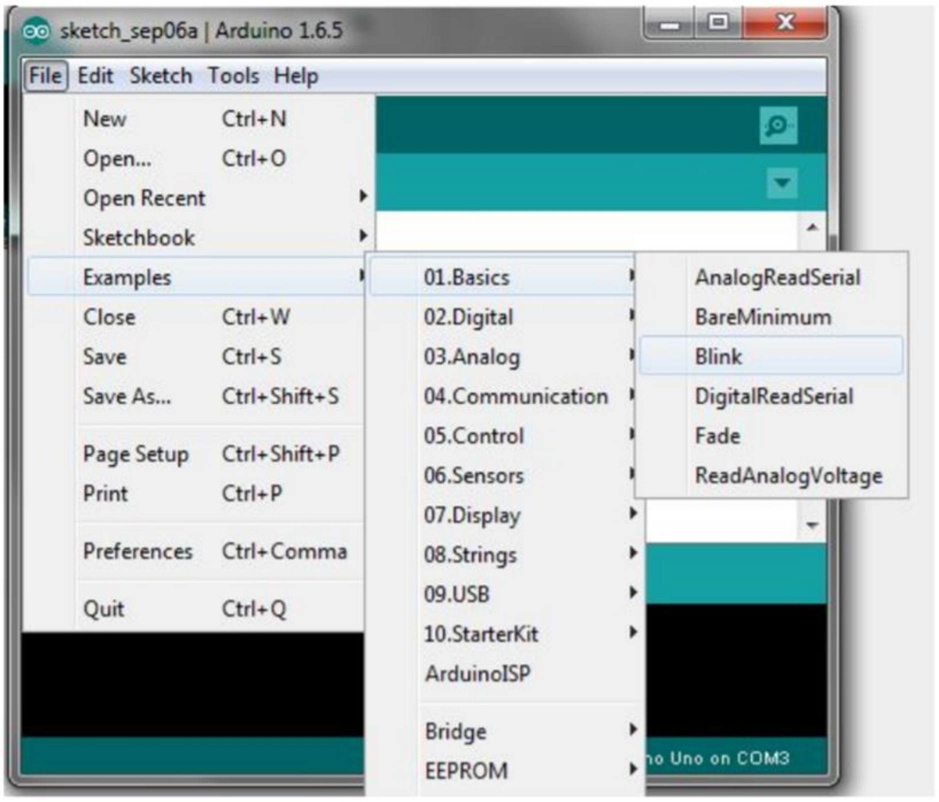
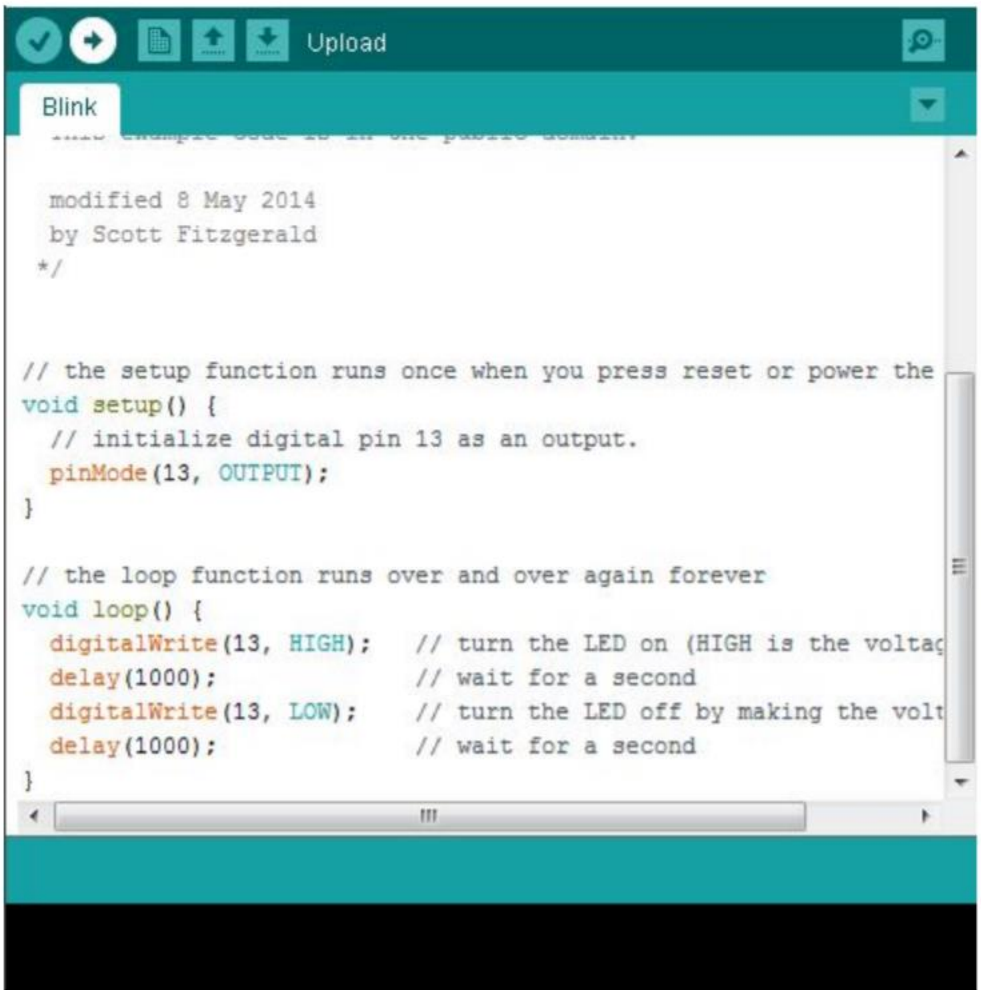


Figure 4.5 Loading Blink Sketch

Fig 4.5 this is a testing code to check if Arduino board is working fine by running blink code



```
 modified 8 May 2014
 by Scott Fitzgerald
*/


// the setup function runs once when you press reset or power the
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltag
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the volt
  delay(1000);              // wait for a second
}
```

Figure 4.6 Blink Code

SKETCH

A sketch is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

A minimal Arduino C/C++ program consist of only two functions:

setup( ): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

loop( ): After setup() function exits (ends), the loop( ) function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Most Arduino boards contain a light-emitting diode (LED) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.[61] A typical program used by beginners, akin to Hello, World!, is "blink", which repeatedly blinks the on-board LED integrated into the

Arduino board. This program uses the functions pinMode( ), digitalWrite()and delay(), which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer.

```
#define LED_PIN 13             // Pin number attached to LED.

void setup() {
   pinMode(LED_PIN, OUTPUT);      // Configure pin 13 to be a digital output.
}

void loop() {
   digitalWrite(LED_PIN, HIGH);   // Turn on the LED.
   delay(1000);                  // Wait 1 second (1000 milliseconds).
   digitalWrite(LED_PIN, LOW);    // Turn off the LED.
   delay(1000);                  // Wait 1 second.
}
```

Libraries

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

Libraries needed

#include <LiquidCrystal.h> //lib for LCD

#include <Servo.h> //Library for Servo Motor

#include <ESP8266WiFi.h> //For Node Mcu ESP8266 Module

```
#include <SPI.h> // Working with LED lights in NodeMcu
```

4.2 SOURCE CODE(IOT)

- **IOT Sample Source Code**

```
// -ve  GnD
// A0 positive
#include <LiquidCrystal.h> //lib for LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int Contrast = 10;

//LCD   Ard-UNO
//pin 1 to gnd
//pin 2 5v
//pin 3 6
//pin 4 12
//pin 5 gnd
//pin 6 11
//pin 11 5
//pin 12 4
//pin 13 3
//pin 14 2
//pin 15 5v
//pin 16 gnd
#include <Servo.h>  //add '<' and '>' before and after servo.h
int servoPin = 8;
 Servo servo;
 int servoAngle = 0;   // servo position in degrees
 #include <SimpleDHT.h>
// for DHT11,
//      VCC: 5V or 3V
//      GND: GND
//      DATA: 2
int pinDHT11 = 7;
SimpleDHT11 dht11(pinDHT11);
#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0
  static unsigned long samplingTime = millis();
#define Offset 0.00 //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40 //times of collection

int pHArray[ArrayLenth]; //Store the average value of the sensor feedback
int pHArrayIndex = 0;
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  Serial.println("pH meter experiment!"); //Test the serial monitor
  servo.attach(servoPin);

  //Lcd Display code Starts Here
  analogWrite(6, Contrast);

  lcd.begin(16, 0);
  lcd.setCursor(0, 1);
  lcd.print(" Weight ");
  lcd.print(" Measurement ");
  delay(1000);
  //lcd.clear();
  //Lcd Display code ends Here
```

```
}
void loop() {
  //PH Sensor Code.....
  static unsigned long printTime = millis();
  static float pHValue, voltage;
  if (millis() - samplingTime > samplingInterval) {
    pHArray[pHArrayIndex++] = analogRead(SensorPin);
    if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
    voltage = avergearray(pHArray, ArrayLenth) * 5.0 / 1024;
    pHValue = 3.5 * voltage + Offset;
    samplingTime = millis();
  }
  if (millis() - printTime > printInterval) {
    Serial.print("Voltage:");
    Serial.print(voltage, 2);
    Serial.print(" pH value: ");
    Serial.println(pHValue, 2);
    digitalWrite(LED, digitalRead(LED) ^ 1);
    printTime = millis();
  }
  //PH sensor code ends here
  // TEMPERATUE sensor start here...
  Serial.println("=================================");
    // read without samples.
  byte temperature = 0;
  byte humidity = 0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err="); Serial.println(err);delay(1000);
    return;
  }
  Serial.print("Temparate and Humidity : -  ");
  Serial.print((int)temperature); Serial.print(" *C, ");
  Serial.print((int)humidity); Serial.println(" H");


  // DHT11 sampling rate is 1HZ.
  delay(1500);
  //TEMP sensor ends here
  // Servo Motor SG90 COde starts here
  //control the servo's direction and the position of the motor


   servo.write(10);      // Turn SG90 servo Left to 45 degrees
   delay(2000);          // Wait 1 second
   servo.write(140);      // Turn SG90 servo back to 90 degrees (center position)
   delay(2000);          // Wait 1 second
//end control the servo's direction and the position of the motor
   //control the servo's speed
   /***
//if you change the delay value (from example change 50 to 10), the speed of the servo
       changes
  for(servoAngle = 0; servoAngle < 180; servoAngle++)  //move the micro servo from
       0 degrees to 180 degrees
  {
    servo.write(servoAngle);
    delay(50);
  }
   for(servoAngle = 180; servoAngle > 0; servoAngle--)  //now move back the micro
       servo from 0 degrees to 180 degrees
  {
    servo.write(servoAngle);
    delay(10);
    }
    //end control the servo's speed
    ***/
    // Servo Motor SG90 Code ends here
  }
  //Manual Function for PH Sensor starts here
  double avergearray(int* arr, int number) {
    int i;
    int max, min;
    double avg;
    long amount = 0;
    if (number <= 0) {
      Serial.println("Error number for the array to avraging!/n");
      return 0;
    }
    if (number < 5) { //less than 5, calculated directly statistics
      for (i = 0; i < number; i++) {
        amount += arr[i];
      }
      avg = amount / number;
      return avg;
    }
    else {
      if (arr[0] < arr[1]) {
        min = arr[0]; max = arr[1];
      }
      else {
        min = arr[1]; max = arr[0];
      }
      for (i = 2; i < number; i++) {
        if (arr[i] < min) {
          amount += min; //arr<min
      min = arr[i];
        }
        else {
          if (arr[i] > max) {
            amount += max; //arr>max
            max = arr[i];
          }
          else {
            amount += arr[i]; //min<=arr<=max
          }
      }
       }
       avg = (double)amount / (number - 2);
       }
      return avg;
     }
            //Manual Function for PH Sensor ends here
```

➢ Admin Registration (HTML)

```
<!DOCTYPE html>
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-database.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
   https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="js/firebase.js">
</script>
<html lang="en">
  <head>
    <meta charset="utf-8">
```

```
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Creative - Bootstrap 3 Responsive Admin
Template">
    <meta name="author" content="GeeksLabs">
    <meta name="keyword" content="Creative, Dashboard, Admin, Template, Theme,
Bootstrap, Responsive, Retina, Minimal">
     <link rel="shortcut icon" href="Img/letter-v.png">
    <title>User Registration - Aqua Manager</title>
    <!-- Bootstrap CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- bootstrap theme -->
    <link href="css/bootstrap-theme.css" rel="stylesheet">
    <!--external css-->
    <!-- font icon -->
    <link href="css/elegant-icons-style.css" rel="stylesheet" />
    <link href="css/font-awesome.min.css" rel="stylesheet" />
    <!-- Custom styles -->
    <link href="css/style.css" rel="stylesheet">
    <link href="css/style-responsive.css" rel="stylesheet" />


    <!-- HTML5 shim and Respond.js IE8 support of HTML5 -->
    <!--[if lt IE 9]>
      <script src="js/html5shiv.js"></script>



      <script src="js/respond.min.js"></script>
      <script src="js/lte-ie7.js"></script>
    <![endif]-->

<style type="text/css">
.form-style-10{
      width:450px;
    padding:30px;
    margin:40px auto;
    background: #FFF;
    border-radius: 10px;
    -webkit-border-radius:10px;
    -moz-border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.13);
    -moz-box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.13);
    -webkit-box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.13);
}
.form-style-10 .inner-wrap{
    padding: 30px;
    background: #F8F8F8;
    border-radius: 6px;
    margin-bottom: 15px;
}
.form-style-10 h1{
    background: #2A88AD;
    padding: 20px 30px 15px 30px;
    margin: -30px -30px 30px -30px;
    border-radius: 10px 10px 0 0;
    -webkit-border-radius: 10px 10px 0 0;
    -moz-border-radius: 10px 10px 0 0;
    color: #fff;
    text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.12);
    font: normal 30px 'Bitter', serif;
    -moz-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    -webkit-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    border: 1px solid #257C9E;
```

```
}
.form-style-10 h1 > span{
    display: block;
    margin-top: 2px;
    font: 13px Arial, Helvetica, sans-serif;
}
.form-style-10 label{
    display: block;
    font: 13px Arial, Helvetica, sans-serif;
    color: #888;
    margin-bottom: 15px;
}
.form-style-10 input[type="text"],
.form-style-10 input[type="date"],
.form-style-10 input[type="datetime"],
.form-style-10 input[type="email"],
.form-style-10 input[type="number"],
.form-style-10 input[type="search"],
.form-style-10 input[type="time"],
.form-style-10 input[type="url"],
.form-style-10 input[type="password"],
.form-style-10 textarea,
.form-style-10 select {
    display: block;
    box-sizing: border-box;
    -webkit-box-sizing: border-box;

    -moz-box-sizing: border-box;
    width: 100%;
    padding: 8px;
    border-radius: 6px;
    -webkit-border-radius:6px;
    -moz-border-radius:6px;
    border: 2px solid #fff;
    box-shadow: inset 0px 1px 1px rgba(0, 0, 0, 0.33);
    -moz-box-shadow: inset 0px 1px 1px rgba(0, 0, 0, 0.33);
    -webkit-box-shadow: inset 0px 1px 1px rgba(0, 0, 0, 0.33);
}

.form-style-10 .section{
    font: normal 20px 'Bitter', serif;
    color: #2A88AD;
    margin-bottom: 5px;
}
.form-style-10 .section span {
    background: #2A88AD;
    padding: 5px 10px 5px 10px;
    position: absolute;
    border-radius: 50%;
    -webkit-border-radius: 50%;
    -moz-border-radius: 50%;
    border: 4px solid #fff;
    font-size: 14px;
    margin-left: -45px;
    color: #fff;
    margin-top: -3px;
}
.form-style-10 input[type="button"],
.form-style-10 input[type="submit"]{

    background: #2A88AD;
    padding: 8px 20px 8px 20px;
    border-radius: 5px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    color: #fff;
    text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.12);
    font: normal 30px 'Bitter', serif;
    -moz-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    -webkit-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.17);
    border: 1px solid #257C9E;
    font-size: 15px;
}
.form-style-10 input[type="button"]:hover,
.form-style-10 input[type="submit"]:hover{
    background: #2A6881;
    -moz-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.28);
    -webkit-box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.28);
    box-shadow: inset 0px 2px 2px 0px rgba(255, 255, 255, 0.28);
}
.form-style-10 .privacy-policy{
    float: right;
    width: 250px;
    font: 12px Arial, Helvetica, sans-serif;
    color: #4D4D4D;
    margin-top: 10px;
    text-align: right;
}
</style>
  </head>
```

```
<body>
<!-- container section start -->
<section id="container" class="">
    <!--header start-->
    <?php include ("headers.php"); ?>
    <!--header end-->


    <!--sidebar start-->
   <?php include ("sidebar.php"); ?>
    <!--main content start-->
    <section id="main-content">
        <section class="wrapper">
                  <div class="row">
                                   <div class="col-lg-12">
                                            <h3 class="page-header"><i class="fa fa-table"></i>User
Registration  </h3>
                                            <ol class="breadcrumb">
                                                     <li><i class="fa fa-home"></i><a
href="index.php">Home</a></li>
                                                     <li><i class="fa fa-plus"></i>User Registration
</li>
                                            </ol>
                                   </div>
                          </div>
              <!-- page start-->


              <div class="row">
                  <div class="col-lg-12">
                      <section class="panel">


                          <div class="table-responsive">
                            <div class="container">
```

```
<div class="row">
      <div class="form-style-10">
<h1>Registration!</h1>
<!--<form action="add_location1.php" method="post" onsubmit="required()>-->
      <div class="section"><span>1</span>Name</div>
    <div class="inner-wrap">
        <label>Name <input type="text" name="registration" id="registration"   required data-
validation-required-message="Please enter registration ID." /></label>
    </div>


   <div class="section"><span>2</span>Email &amp; Phone</div>
    <div class="inner-wrap">
        <label>Email Address <input type="email" name="email"  id="email" required data-
validation-required-message="Please enter your Email Address." /></label>
        <label>Phone Number <input type="text" name="phone" id="phone"
pattern="[789][0-9]{9}" maxlength="10" required data-validation-required-
message="Please enter your Phone Number."/></label>
    </div>


      <div class="section"><span>3</span>Password</div>
        <div class="inner-wrap">


        <label>Password <input type="text" id="pass" name="pass" required data-validation-
required-message="Please enter your Password" /></label>


    </div>


    <div class="section"><span>4</span>Aquarium ID</div>
        <div class="inner-wrap">



 <label>Aquarium ID <input type="text" id="aqua" name="aqua" required data-validation-
required-message="Please enter Aquarium ID" /></label>
```

```
   </div>
  <div class="button-section">
   <input type="submit" id="addBtn" onclick="myFunction()" />
   <span class="privacy-policy">
   <!--<input type="checkbox" name="field7">You agree to our Terms and Policy. -->
   </span>
  </div>
<!--</form>-->
</div>
</div>
                </div>
                        </div>

                    </section>
                </div>
            </div>
            <!-- page end-->
        </section>
    </section>
    <!--main content end-->
  </section>
  <!-- container section end -->
   <!-- javascripts -->
   <script src="js/jquery.js"></script>
   <script src="js/bootstrap.min.js"></script>

   <!-- nicescroll -->

   <script src="js/jquery.scrollTo.min.js"></script>
   <script src="js/jquery.nicescroll.js" type="text/javascript"></script>
   <!--custome script for all page-->
```

```
    <script src="js/scripts.js"></script>
    <script src="js/function.js"></script>


    <!--data insertion-->
    <!-- <script>alert("'registration' USER SUCCESSFULLY ADDED TO THE
DATABASE.");window.location="index.php"</script> -->


<script>
function myFunction() {
    alert("REGISTRATION USER SUCCESSFULLY ADDED TO THE
DATABASE.");window.location="index.php"
  //alert("I am an alert box!");
}
</script>


  </body>
</html>
```

- Analysis of data(HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Creative - Bootstrap 3 Responsive Admin Template">
    <meta name="author" content="GeeksLabs">
    <meta name="keyword" content="Hungers - Admin">
    <link rel="shortcut icon" href="Img/letter-v.png">
    <script src="https://www.gstatic.com/firebasejs/3.3.0/firebase.js"></script>
    <title>Aqua Manager - Admin</title>


    <!-- Bootstrap CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- bootstrap theme -->
    <link href="css/bootstrap-theme.css" rel="stylesheet">
    <!--external css-->
    <!-- font icon -->
    <link href="css/elegant-icons-style.css" rel="stylesheet" />
    <link href="css/font-awesome.min.css" rel="stylesheet" />
    <!-- full calendar css-->
    <link href="assets/fullcalendar/fullcalendar/bootstrap-fullcalendar.css" rel="stylesheet" />
        <link href="assets/fullcalendar/fullcalendar/fullcalendar.css" rel="stylesheet" />
    <!-- easy pie chart-->
    <link href="assets/jquery-easy-pie-chart/jquery.easy-pie-chart.css" rel="stylesheet"
type="text/css" media="screen"/>
    <!-- owl carousel -->
    <link rel="stylesheet" href="css/owl.carousel.css" type="text/css">
        <link href="css/jquery-jvectormap-1.2.2.css" rel="stylesheet">
    <!-- Custom styles -->
        <link rel="stylesheet" href="css/fullcalendar.css">
        <link href="css/widgets.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
    <link href="css/style-responsive.css" rel="stylesheet" />
        <link href="css/xcharts.min.css" rel=" stylesheet">
        <link href="css/jquery-ui-1.10.4.min.css" rel="stylesheet">
  </head>


  <body>
  <!-- container section start -->
  <section id="container" class="">
```

```
<!--header end-->
<?php include ("headers.php");  ?>

<!--sidebar start-->
 <?php include ("sidebar.php"); ?>
<!--sidebar end-->

<!--main content start-->
<section id="main-content">
   <section class="wrapper">
      <!--overview start-->
                        <div class="row">
                                <div class="col-lg-12">
                                        <h3 class="page-header"><i class="fa fa-laptop"></i>
Dashboard</h3>
                                        <ol class="breadcrumb">
                                                <li><i class="fa fa-home"></i><a
href="index.php">Home</a></li>
                                                <li><i class="fa fa-laptop"></i>Dashboard </li>

                                        </ol>
                                </div>
                        </div>


    <div class="row">
                                <div class="col-lg-3 col-md-3 col-sm-12 col-xs-12">
   <a href="user_list.php">
                                        <div class="info-box blue-bg">
                                                <i class="fa fa-user" aria-hidden="true"></i>
                                                <div class="count"></div>
 <div class="title">User</div>
                                              </div><!--/.info-box-->
                                  </a></div><!--/.col-->



                                  <div class="col-lg-3 col-md-3 col-sm-12 col-xs-12">
<a href="https://community.thingspeak.com/forum/mobile-apps/">
                                              <div class="info-box red-bg">
                                                          <i class="fa fa-file"></i>
                                                          <div class="count"></div>
                                                          <div class="title">Complaint</div>

                                              </div><!--/.info-box-->
                                  </div><!--/.col-->


            </div><!--/.row-->


   <!- List starts -->
                                  <ul class="today-datas">
     <!-- List #1 -->
                                  <li>
       <!-- Graph -->
       <div><span id="todayspark1" class="spark"></span></div>
       <!-- Text -->
       <div class="datas-text"> Logins - Today</div>
     </li>
     <li>
       <div><span id="todayspark2" class="spark"></span></div>
       <div class="datas-text">Newsletter - Today</div>
     </li>
     <li>
```

```
         <div><span id="todayspark3" class="spark"></span></div>
             <div class="datas-text"> Contact - Today</div>
           </li>

           <!--<li>
             <div><span id="todayspark5" class="spark"></span></div>
             <div class="datas-text">12,000000 Orders - Today</div>
           </li>    -->
          </ul>
          </div>
        </div>
          <!-- project team & activity start -->
    <br><br>

          <!-- project team & activity end -->

       </section>
       <div class="text-right">
      </div>
    </section>
    <!--main content end-->
</section>
<!-- container section start -->

  <!-- javascripts -->
  <script src="js/jquery.js"></script>
        <script src="js/jquery-ui-1.10.4.min.js"></script>
  <script src="js/jquery-1.8.3.min.js"></script>
  <script type="text/javascript" src="js/jquery-ui-1.9.2.custom.min.js"></script>
  <!-- bootstrap -->
  <script src="js/bootstrap.min.js"></script>
```

```
   <!-- nice scroll -->
   <script src="js/jquery.scrollTo.min.js"></script>
   <script src="js/jquery.nicescroll.js" type="text/javascript"></script>
   <!-- charts scripts -->
   <script src="assets/jquery-knob/js/jquery.knob.js"></script>
   <script src="js/jquery.sparkline.js" type="text/javascript"></script>
   <script src="assets/jquery-easy-pie-chart/jquery.easy-pie-chart.js"></script>
   <script src="js/owl.carousel.js" ></script>
   <!-- jQuery full calendar -->
   <<script src="js/fullcalendar.min.js"></script> <!-- Full Google Calendar - Calendar
-->
        <script src="assets/fullcalendar/fullcalendar/fullcalendar.js"></script>
   <!--script for this page only-->
   <script src="js/calendar-custom.js"></script>
        <script src="js/jquery.rateit.min.js"></script>
   <!-- custom select -->
   <script src="js/jquery.customSelect.min.js" ></script>
        <script src="assets/chart-master/Chart.js"></script>


   <!--custome script for all page-->
   <script src="js/scripts.js"></script>
   <!-- custom script for this page-->
   <script src="js/sparkline-chart.js"></script>
   <script src="js/easy-pie-chart.js"></script>
        <script src="js/jquery-jvectormap-1.2.2.min.js"></script>
        <script src="js/jquery-jvectormap-world-mill-en.js"></script>
        <script src="js/xcharts.min.js"></script>
        <script src="js/jquery.autosize.min.js"></script>
        <script src="js/jquery.placeholder.min.js"></script>
        <script src="js/gdp-data.js"></script>
        <script src="js/morris.min.js"></script>
        <script src="js/sparklines.js"></script>
        <script src="js/charts.js"></script>
        <script src="js/jquery.slimscroll.min.js"></script>
<script>

  //knob
  $(function() {
    $(".knob").knob({
      'draw' : function () {
        $(this.i).val(this.cv + '%')
      }
    })
  });

  //carousel
  $(document).ready(function() {
      $("#owl-slider").owlCarousel({
          navigation : true,
          slideSpeed : 300,
          paginationSpeed : 400,
          singleItem : true

      });
  });

  //custom select box

  $(function(){
      $('select.styled').customSelect();
  });
          /* ---------- Map ---------- */
```

```
$(function(){
  $('#map').vectorMap({
    map: 'world_mill_en',
    series: {
      regions: [{
        values: gdpData,
        scale: ['#000', '#000'],
        normalizeFunction: 'polynomial'
      }]
    },
            backgroundColor: '#eef3f7',
    onLabelShow: function(e, el, code){
      el.html(el.html()+' (GDP - '+gdpData[code]+')');
    }
  });
});

  </script>

  </body>
</html>
```

➢ Datacollection(Android)

```
package com.example.eventmanager;
import android.app.ProgressDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ProgressBar;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class DataActivityextends Fragment implements
    Dashboard_adapter.ClickAdapterListener {
  View v;
```

```
private RecyclerView recyclerView;
private LinearLayoutManager linearLayoutManager;
private ArrayList<Dashboard_item_model> modelist;
private Dashboard_adapter dashboard_adapter;
private Context ctx;
private BroadcastReceiver MyReceiver = null;

private DatabaseReference databaseReference;
//private FirebaseMethods firebaseMethods;
private int imagecount = 0;
private ProgressDialog mProgress;

@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
    container, @Nullable Bundle savedInstanceState) {
    v = inflater.inflate(R.layout.events, container, false);

    linearLayoutManager = new LinearLayoutManager(getActivity());
    recyclerView = v.findViewById(R.id.recycler);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(linearLayoutManager);



    databaseReference = FirebaseDatabase.getInstance().getReference("event");
    //final String single_view = getRef(position).getKey();
//progress bar
    mProgress = new ProgressDialog(getActivity(),
     R.style.Theme_AppCompat_DayNight_Dialog_Alert);
    mProgress.setIndeterminate(true);
    //mProgress.setTitle("Processing...");
```

```
mProgress.setMessage("Loading");
mProgress.setCancelable(false);
mProgress.show();


modelist = new ArrayList<>();
databaseReference.addValueEventListener(new ValueEventListener() {
   @Override
   public void onDataChange(@NonNull DataSnapshot dataSnapshot) {


      modelist.clear();


      for (DataSnapshot dataSnapshot1 : dataSnapshot.getChildren()) {


         Dashboard_item_model modl = dataSnapshot1.getValue(Dashboard_item_model.class);
         modelist.add(modl);
      }


      dashboard_adapter = new Dashboard_adapter(getActivity(), modelist, Events.this);
      recyclerView.setAdapter(dashboard_adapter);
      mProgress.dismiss();
     /* Asyncprogress T = new Asyncprogress(getContext());
      T.execute();   // this will call do in background
*/
      /*ItemTouchHelper.SimpleCallback itemTouchHelperCallback = new RecyclerItemTouchHelper(0, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT , Dashboard_Frag1.this);
      new ItemTouchHelper(itemTouchHelperCallback).attachToRecyclerView(recyclerView);
      ItemTouchHelper.SimpleCallback itemTouchHelperCallback1 = new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT | ItemTouchHelper.UP) {
            @Override
            public boolean onMove(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder, RecyclerView.ViewHolder target) {
                return false;
            }

            @Override
            public void onSwiped(RecyclerView.ViewHolder viewHolder, int direction) {
                // Row is swiped from recycler view
                // remove it from adapter
            }

            @Override
            public void onChildDraw(Canvas c, RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder, float dX, float dY, int actionState, boolean isCurrentlyActive) {
                super.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);
            }
        };

        // attaching the touch helper to recycler view
        new ItemTouchHelper(itemTouchHelperCallback1).attachToRecyclerView(recyclerView);*/
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
        Toast.makeText(getContext(), "Error....!!", Toast.LENGTH_SHORT).show();

        /*Asyncprogress T = new Asyncprogress(getContext());
        T.execute();   // this will call do in backgroun*/
```

```
            }
        });

        return v;

    }



    public void onBackPressed() {
    }

    @Override
    public void onRowClicked(int position, View v) {
        Intent intent = new Intent(getActivity(), Bookactivity.class);
        //intent.putExtra("Eventname",modelist.get(position).getEvent_name());
        intent.putExtra("event_name", modelist.get(position).getEvent_name());
        // intent.putExtra("center_name", modelist.get(position).getCenter_name());
        intent.putExtra("date", modelist.get(position).getDate());
        intent.putExtra("imageView", modelist.get(position).getImageView());
        intent.putExtra("stime", modelist.get(position).getStime());
        intent.putExtra("price", modelist.get(position).getPrice());
        /*imageView.buildDrawingCache();
        Bitmap bitmap = imageView.getDrawingCache();
        intent.putExtra("BitmapImage", bitmap);
*/
        startActivity(intent);
        //getActivity().finish();
    }
      }
```

4.3 TESTING ENVIRONMENT

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured.

Test bed or test environment is configured as per the need of the Application Under Test. On a few occasion, test bed could be the combination of the test environment and the test data it operates.

Setting up a right test environment ensures software testing success. Any flaws in this process may lead to extra cost and time to the client.

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

Factors for designing Test Environment

- Determine if test environment needs archiving in order to take back-ups.
- Verify the network configuration.
- Identify the required server operating system, databases and other components.
- Identify the number of license required by the test team.

Fish are "cold-blooded" and therefore assume the temperature of the water they live in. Water temperature is therefore the most important physical factor for fish survival and growth. Body temperature, and thus the water temperature, has an effect on level of activity, behavior, feeding, growth, and reproduction of the fish. Each species has its tolerance limits and optimum range. When water temperatures are outside the optimum range, fish body temperature will either be too high or too low and fish growth will be affected or the fish will even die.

Water quality gets affected due to many reasons such as fish producing waste, oily surface due to food given to the fishes etc. To identify this unhealthy environment with naked eye is very difficult.

Our project test different condition of water, humidity and temperature every 20 second. If any changes are found user gets a notification like change water, provide proper humidity and temperature for the fishes. It is very important to maintain a healthy environment for the fishes inside an aquarium. Hence the quality of water and humidity and temperature is being tested for following condition: -

- Ensure fishes are getting adequate humidity and temperature.
- Check water provided for fishes is safe or not.

We should take some factors in consideration-

- Check from where water is taken for filling aquarium for the fishes.
- Turn on/off the light whenever temperature is required.
- In case of high temperature or humidity decrease the intensity of light
- In case of low temperature or humidity increase the intensity of light
- To ensure the ammonia and nitrite levels have not reached harmful levels

4.4Test Plan

Test Plan is a dynamic document. The success of a testing project depends upon a well-written test plan document that is current at all times. Test Plan is more or less like a blueprint of how the testing activity is going to take place in a project.

The objectives of testing include-

- To make sure that the pH sensor electrode is working fine and the pH values of the water are getting determined.
- To determine the correct pH values of the freshwater as the determination of water quality depends on the levels of the pH value of the water.
- To ensure that pH value of the water for freshwater lies between the range of 6.8 to 7.6 as the standard scale of pH is from 0-14.
- To determine DHT11 sensor is able to detect humidity and temperature inside an aquarium.
- LED light will determine whether nodeMCU is connected with server or not.
- To determine whether servo motor’s motor is working smooth

5. RESULTS AND DISCUSSION

6. CONCLUSION

REFERENCES

[1]. "Introducing the MySQL 8 Document Store"
url: https://www.apress.com/gp/book/9781484227244

[2]. “MySQL 8.0 Reference Manual” url:
“https://downloads.mysql.com/docs/refman-8.0-en.pdf “

[3]. Patrick Dalton, Paul Whitehead, “SQL SERVER BLACK BOOK”
Dream Tech New Delhi, 2005.

[4]. Roger Pressman, “Software Engineering A Practitioner’s Approach”,
McGraw-Hill Company, 2001.