SmartZoo Controlling & Monitoring System

California State University, Fresno
Electrical and Computer Engineering Department
ECE 186B - Senior Design II
Spring 2023 - Dr. Tayeb
Technical Advisor: Dr. Tayeb

Group Members:
Puya Fard

Dominic Keifer

Sahildeep Singh

Rafael Hernandez

REVISION HISTORY

Version Number	Advisor Comments	Member Name	Date	Changes made
1.0	N/A	Group	10/26/22	Initial setup for the document.
1.1	N/A	Puya Fard	10/29/22	Added BibTex.
1.2	N/A	Puya Fard	10/30/22	Structure and flow
1.3	N/A	Puya Fard	11/21/22	Figures
1.4	N/A	Puya Fard	11/28/22	Abstract and gantt chart
1.5	N/A	Dominic Keifer	12/11/22	Grammarand and syntax
1.6	N/A	Group	12/16/22	Finalize the charter
1.7	N/A	Puya Fard	3/5/2023	Update progress made
1.8	N/A	Puya Fard	3/13/2023	Abstract, introduction, block diagrams
1.9	N/A	Puya Fard	4/23/2023	Peer-reviewed final report
1.10	N/A	Puya Fard	5/6/2023	Quantitive analysis

ABSTRACT

This project is collaborative teamwork among three computers and one electrical engineer, and it's called SmartZoo Controlling & Monitoring system. Its main objective is to create a safe and smart control environment that assures sensitive animals' safety in the zoo. This project utilizes embedded systems, cloud-based programming, and a software-based user interface to monitor, collect, and store data. The collected data gets utilized to prevent potential safety hazards by providing insights into the environment's temperature, humidity, ph-level, and oxidation level. There is also a camera module to monitor the environment of the animal for other potential safety hazards by unauthorized humans. This project includes solar module design to provide clean energy production for the system. As for dataflow, collected data is stored in a collective database in Firebase cloud, utilizing machine learning to predict water and air temperature, and resulting variables from that data collection are monitorable via the ”smartzoo.net” website, which also offers accessibility to permitted users who can monitor the data and send actuator input feedback to change the parameter of a variable. Furthermore, this project conducted a detailed power and quantitive analysis to answer any questions regarding performance measurements, power consumption, and delays that are presented in the overall and individual components of the system. In summary, this project assists zoo workers and the zoo itself by delivering a smart, controlling, and monitoring environment that protects sensitive zoo animals. Each smart system cost nearly $450 to build, depending on the quality of the sensors used.

TABLE OF CONTENTS

CONTENTS

LIST OF FIGURES

List of Tables

I. Introduction

This project is going to be a smart environmental control system that will monitor critical measurements that are necessary for sensitive zoo animals' ecosystems and habitats. Depending on the acquired and stored data from measurements, the system is going to adjust parameters either systematically or with a user input control to maintain the delicate environments that animals are required to be in for their safety and well-being. The system utilizes an embedded system, which is a microcontroller that will be responsible for collecting data using its connected sensors, cloud-based programming using Firebase cloud packages, which contains all the data transferred from our microcontroller and stored for up to two days, and finally a user-interface website to monitor the data stored in our cloud storage. The purpose of this project is to allow zookeepers to see the current environment details of the animal where the node is configured as well as allow them to make changes with a help of an actuator. However, this system is also going to be configured to take critical actions if required if human input is delayed or forgotten.

Many animals in zoos, especially the ones that live in water, are required to be kept in a very tight set of environmental parameters, such as temperature level, oxygen level, ph-level, humidity, and cleanness that make up water in their environments. This project monitors and controls the living standards of stingray animals via temperature, ph-level, and oxidation-level sensors to insure the safety of their water tank. The second environment and animal group is elephant water and the importance behind this comes from elephants that are kept in zoos are limited to a single water park in their lockdown area. Therefore, they drink the same water that they bathe and potentially excrement in, so it's important to check the cleanness of the water so that they don't get sick and develop health problems that could potentially risk their lives. The sensors that will be used to provide cleanness feedback from their water park are temperature, ph-level, and oxidation level. Finally, our third animal group is reptiles. Certain reptiles are required to be kept under a certain range of temperature and humidity for their well-being. Therefore, our project will utilize humidity and temperature sensor to insure the quality of the reptiles' environment for their well-being. It's important to note that all these animals are located in Fresno Chafee Zoo, and this project is a potential advocate to be used in the future by them for the animals listed above.

A little about the specifics of our project, the microcontroller used for this project are Arduino UNO Rev-2, ESP8266 Wrover D1 R1, and ESP32 microcontroller as a camera module. Arduino UNO rev-2 is powered with the help of a solar module, ESP8266 Wrover D1 R1 is powered via its AC/DC charger, and ESP32 has powered via its USB-c charger. Individual power analyses of each module have been conducted in this project and could be found under their own subsection of this report. The sensors are budget-friendly since this is a senior design project made by students, however, it could be designed with more advanced sensors that deliver more accurate data with additional features that could potentially be used to add more complexity and safety measurements for animal habitats. The cloud system that is used in this project is powered via Firebase, which provides all aspects required for our project to deliver a full working prototype, it includes machine learning, authentication user login system, data, and image storage, and its own built-in website template. And finally, the user interface is a website that is also mobile screen friendly, which will allow users to log in to their portal and view data stored within certain time periods.

Within this team, Puya Fard is responsible for being the project manager, researching and completing some of the sensors that are going to be connected to our microcontroller. He will configure these sensors to their working, data storing, and communicating states. Dominic Keifer is responsible for completing the Camera module that will contain an AI motion detector as well as establishing wireless communication between the microcontroller-cloud-internet, and data server, and participating in the creation of website applications. Sahildeep Singh is responsible for the design of the website application, assisting with communication protocols, cloud functions and programming, and helping Puya with the WiFi module as well as the timekeeping required for our project. And finally, Rafael Hernandez is responsible for sensor connections and interface, Solar module, and helping Puya with sensors as well as scheduling and organization. Since we are a four-member team, we made sure our project's complexity is good enough to satisfy the rubric by meeting with our course instructor and technical advisor Dr. Shahab Tayeb [1].

II. Project Objectives and Success Criteria

A. *Project objective*

This project's objective is to create a safe and sustainable environment for sensitive zoo animals such as Stingrays [2], Elephants [3], and Reptiles [4]. This project is going to benefit zookeepers and the zoo itself by digitalizing the monitoring and controlling tasks of these animals and leaving human error aside. Zookeepers will benefit from this project by accessing the nodes via WiFi and monitor & controlling it on demand. The zoo will benefit from this project by having their zookeepers at other tasks necessary that could be multitasked with monitor& controlling tasks. Our success criteria are to have our node do the required measurements, required data storing, and required cloud-based communication with a user behind a website that will be able to monitor, control, and judge the parameters acquired from the node. To accomplish this project, we have divided the project into many tasks and subtasks that will be put together as a group and complete the end product, a smart control system.

B. *Success criteria*

This project is required to deliver a full working prototype for the tasks listed. It must collect accurate data from its sensors, deliver data to Firebase and store up to two days of data, and allow users to monitor data within its specified time period, as well as provide instantaneous data when requested. It must also do action regarding one of the collected variables, in this case the temperature of the environment, in a critical situation via user feedback sent from the website to the microcontroller, which will satisfy the controlling part of this project.

III. High-Level Requirements

Our project can be divided into seven sections that will help us visualize the high-level requirements. The first section is the configuration of each sensor and node as well as adjusting their parameters and data storage. This section will be completed by the end of this Fall 2022 semester to serve the purpose of prerequisite for the start of the next Spring 2023 semester and ECE 186B. The second section is the configuration of the Camera module and AI motion detector which will also be finished by the end of this semester and serve the purpose of prerequisite. The third section is to configure and complete the power consumption of the device and node with both AC/DC sources along with the Solar module that will be implemented by our Electrical Engineer team member Rafael by the beginning of ECE 186B. The fourth section comes with a prerequisite of all the sections before to be completed successful test runs and trials. This section will establish WiFi communication between the cloud and the microcontroller's collected data. This section will focus on how the data is stored in the cloud and how it will be communicated along with its own timing constraints and data flow rate. The fifth section is going to deal with user-interface application implementation over websites. In this section, our assigned team members will learn different coding styles and languages that website applications require to be functioning. The sixth section is to establish successful communication between the user-interface application and the cloud as well as make sure the data input from the user is registered in the cloud and transferred. The seventh and final section is to establish a successful two-way communication between the user interface and microcontroller via WiFi and cloud and run some tests which will result in our End product.

IV. Assumptions, Constraints, and Standards

A. Team members background information

Our team consists of Puya Fard, Dominic Keifer, Rafael Hernandez, and Sahildeep Singh. Puya, Dominic, and Sahildeep are all Computer Engineers while Rafael is an Electrical Engineer. Dominic is an employee at Fresno Chaffee Zoo who works during night shifts and after visitor hours. Dominic is responsible for monitoring and checking sensitive zoo animals' exhibit parameters every couple of hours to make sure there are no critical changes in their ecosystem that might harm them. Therefore, he has valuable insight into the zoo's animal ecosystems which makes it a great source of information that we can take advantage of when it comes to learning more about stingray animals, elephants, and reptiles that will be used for this project. Another advantage we have on is that this project is that most of us are Computer Engineers, so any tasks that require programming can be done with broad course knowledge since a good portion of our project will be from similar ECE courses as our major requirement. During the first couple weeks of the ECE 186A course, our instructor Dr. Tayeb has given us a Clifton strengths and weaknesses assessment [5] to evaluate our individual strengths and weaknesses. Puya, our group leader reached out to the rest of the team members with correlating and satisfying strengths and weaknesses, then formed the group. Our group believes that the mixture of our strengths and weaknesses formed a successful communication and working group along with a strong leader.

B. Technical faculty advisor

Our technical faculty advisor is Dr. Shahab Tayeb [1] who has expertise in wireless sensor networks. We plan on using his knowledge of wireless sensor networks as well as cloud-based programming and its application when it comes to establishing successful communication between the user and node end to transfer the data collected and stored. Since he is also our course instructor, we will further use his expertise on our physical device configuration to make sure we are on the most optimal path when it comes to data storage and communication with the device itself. And finally, we plan on getting help with the networking of the user-application design part of our project which is responsible for monitoring and controlling the data coming from cloud-based communication.

C. Courses used as resources

Our group consists of three computer engineering majors and one electrical engineer, which makes us have solid background knowledge in areas of programming and implementation. Some of these courses are such as ECE 85 Digital Logic Design, ECE 90 Principals of Electrical Circuits, and ECE 128 Electronics which are general ECE courses that we all have completed before and are ready to assist us with the general purpose and tasks such as sensor configuration and connectivity of this project. Some topics that will be used from ECE 85, 90, and 128 courses are utilizing logic gates, use of basic electrical instruments while implementing the physical layer of our project, verification of basic circuit laws and principles along with their lab equipment to measure voltage and power supply design for solar power consumption. Moreover, ECE 118 Microprocessor Architecture and Programming, ECE 176 Advanced Digital Logic Design, and ECE 178 Embedded Systems will assist us with applications of this project that deal with data processing and collection, parallel and serial I/O communications, CPU utilization, data transfer scheduling, and timing constraints of data communication of this project [6]. And finally, CSCI 40 Intro to engineering applications and CSCI 41 Data Structure courses is going to assist us with the coding and user interface side of the project.

D. Background: Team members' previous projects

All of our team members have conducted many projects throughout their previous years of studying. These projects taught us to visualize the big picture along with figuring out possible routes for completion.

Puya's projects and coursework that will be applicable to this course consist of a music synthesis program in MATLAB from ECE 72, a game on an FPGA board in ARM Assembly from ECE 118, a number guessing game in ARM Assembly from ECE 118 lab, an IoT project for vehicles from ECE 146, a parallel to serial interface using synchronous FIFO buffer system in Verilog from ECE 176 and implementing a Netflix menu hub in C++ from CSCI 41.

Dominic's projects and coursework that will be applicable to this course consist of creating a game on the DE1 board from ECE 118, an IoT smart city project from ECE 146, a keylogger from ECE 150, RSA Encryption and Decryption implementation from ECE 156, a DE2-115 FPGA board based self-driving car with sensors, and a bike speedometer CMOS schematic/design from ECE 140.

Rafael's projects and coursework that will be applicable to this course consist of a game designed on the DE1 board for ECE 118 and ECE 118L, designing a schematic and layout for a standard ALU using CMOS. Since he is the only electrical Engineer on our team, his work from courses that focus on circuits will be very useful to us when figuring out how to connect our various components with each other.

Sahil's projects and coursework that will be applicable to this course consist of a snake game in C++ from ECE 71, tail lights using logic gates from ECE 85, implementing a banking system using the DE1 board in ARM Assembly from ECE 118, pipelined ALU from ECE 174, floating point multiplier from ECE 176, improving sorting algorithms in C++ from ECE 141, medical IoT implementation in Cisco Packet tracer from ECE 146, and designing a bike speedometer CMOS schematic.

A lot of our team members have taken similar classes so we have similar expertise on some of the topics. All of our team members have experience in multiple coding languages from projects in courses such as ECE 71, ECE 72, ECE 118, ECE 141, ECE 176, and CSCI 41 to name a few. Our project will have a lot of software involved for things such as our cloud database, app, and website. Therefore, knowing multiple languages will also make learning any new coding languages we need much easier because most follow a similar pattern of how to implement ideas into code. For our projects, some of the most useful projects will be from ECE 146 which most of our team members have taken. From ECE 146 Puya worked on a vehicle IoT system, Dominic worked on a smart city IoT system, and Sahildeep worked on a medical IoT system. The reason why it will be some of the most useful projects is that our senior design project at its core is an IoT system and having knowledge of this topic, such as what IoT model to follow, will help us greatly in creating our IoT system for a zoo.

E. Components background information

The topics we will mainly need to research and gain insight into are the different types of microcontrollers, sensors, cloud services, and application interfaces that are available for us to store, manipulate, and communicate data between our user-cloud-node. For microcontrollers, we researched Arduino UNO, ESP32, ESP8266, Raspberry Pi, Texas Instruments, and Intel Edison boards. For our sensor nodes, we are currently looking at sensors that measure Ph-levels, Temperature levels, Humidity , and Oxidation levels inside water, a Power sensor for all the electronics, and a camera module that will provide a live feed and motion detection. For our cloud service provider, we are looking at Firebase packages that offer free services up to limited Data and time that is more than satisfactory for our project. And finally, for our user-interface applications, we are going to conduct deeper research on how to implement a website application that will allow our users to monitor and control the data collected from our sensors. Other minor components that have completed their individual research for a number of breadboards, jumper wires, alligator clips, USB cables, and enclosed containers for individual modules are what we will need. The correct cables for the WiFi module and the USB the module consists of USB-A and Micro USB. Furthermore, there might be a need for an external storage for storing an excessive amount of the data collected. Finally, the implementation of our actuators that we will need to make the controlling part of the project possible consists of a water proof container, two water heaters, and an IoT power relay to be controlled via cloud programming and timers.

F. Zoo animals background information

In this project, we have conducted background research on three specific zoo animals to provide smart system support for the specifics of their habitat:

i) Stingrays are sensitive water animals that require a specific range of temperature for the water level along with Ph, salinity, and specific gravity level. Some specifics of Stingray animals are: Having a water temperature of between 75-82 F°. Ph level between 6.8 - 7.6 and alkalinity between 1° and 4° (18 ppm to 70 ppm). The salinity level of 25-43 and specific gravity of 1.020-1.025 [2].

ii) Elephants at Fresno zoo bathe themselves once every week in the water pool provided at their habitat. However, the problem starts when they decide to drink the same water they bathed in. Fresno Chaffee zoo is already taking great levels of measurements by monitoring water's Ph-level, oxidation level, and cleanliness by hiring more than a handful of zookeepers. We wanted to provide this system and configure our parameters to decent specifics that their water requires to be in and alert them when they are at critical levels [3].

iii) Reptiles species located at Fresno Chaffee zoo also require to be kept under certain temperature and humidity levels. This is critical when it comes to their survival. This is the final animal species we are including in our project to detect its critical levels and alert our zookeepers via our system [4].

G. Background research conducted at zoo

Our group member Dominic was able to talk to a couple of zookeepers and life support services lead to determine what sensors are required to monitor and maintain an animal such as a stingray [7]. They discussed that the more important or critical sensors are the sensors for Ozone, Oxidation, ORP, temperature, dissolved oxygen, and power consumption by the pumps. They also discussed sensors or levels that are not life- threatening but are still checked and those were levels for PH, NH3(Ammonia), NO2(Nitrogen Dioxide), NO3(Nitrate), and phosphorus. It was also discussed that measuring the difference in pressure between pumps is also useful and all of these sensors need to be able to tolerate water [7].

H. Standards

Our Project will be following a general IoT reference model used by many other IoT applications. This contains four standard layers shown in **Fig. 1**. The first one is the Application layer which will include our software part of the project including a website user interface. Second, the Service layer will contain cloud machine learning and data storage and sorting processes of our project. Third, the Networking layer will include steps and standards that our group needs to follow to establish successful communication between the cloud and microcontroller for successful data transfer. And finally, the Device layer will include all the hardware design and configuration parts of this project, which includes sensors, breadboard, wires, solar panels, batteries, actuators, and microcontroller. Since we are under the ECE department, we thought it would be best to follow this standard.

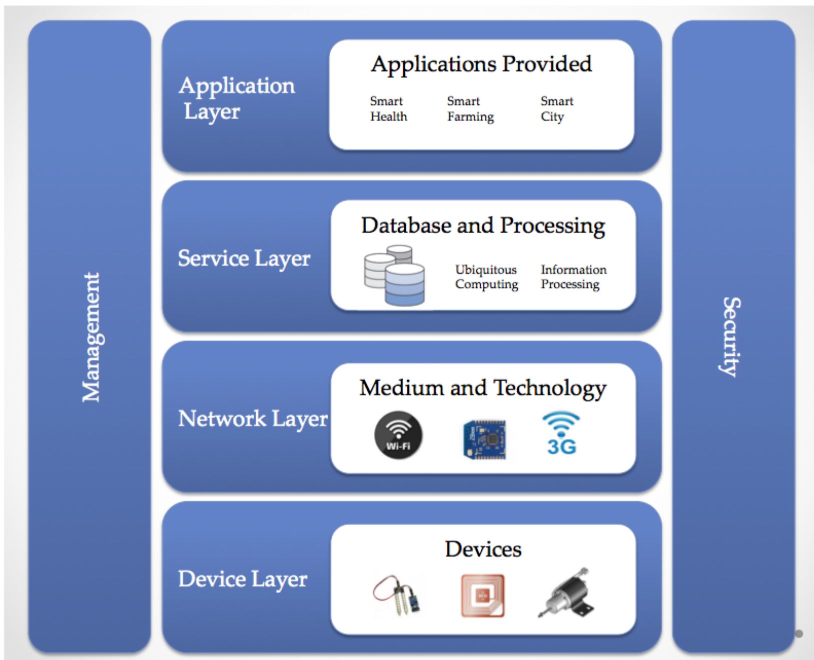


Fig. 1: IoT reference model

The **Fig 1: IoT model** attached above provides detailed information about the reference model our team used for this project. We have a total of four layers, and each layer represents various tasks and checkpoints for themselves to reach the final deliverable product. These layers are referenced from the IoT Architecture model provided and modified to be compatible with our system [8].

But when it comes to modeling these standards to this project's specifics, at the very top we have the application layer which is going to be our user-interface website that will display our collected and stored data from our Firebase cloud storage. In this layer, we provide a secure authentication system that will satisfy our system's security constraints. The service layer is where our Firebase cloud storage and machine learning programming come into play. In this layer, we store data collected from our microcontroller for up to two days and also send the most recent collected data to our users to view from their user-interface website. As mentioned before, machine learning programming is included in this layer to provide predictions of the upcoming day's water temperature for stingrays' water tanks to help with monitoring. The networking layer is going to be WiFi in our instance. We deliver our data thru 2.4 Ghz frequency to our Firebase cloud and communicate with the user. Finally, our device layer is going to be all of our physical components including microcontrollers, solar pannel, powering cables, batteries, sensors, jumper wires, breadboards, and containers to keep everything secure and together. All of these could be visualized under **Fig 2.**.

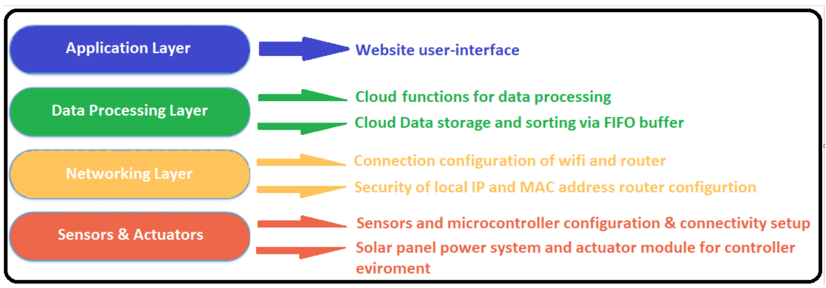


Fig. 2: Modified IoT model

V. Project Description and Boundaries

The Smart Zoo is a project that is going to be a primary control and monitoring system that will have sensors and nodes connected to its microcontroller end, and user input and application interface to the other. And all of these systems will be working and functioning via WiFi and cloud-based programming and communication. A very basic idea behind our project is modeled in **Fig. 3**. As we can see we have three major modules in this project. The software user interface, cloud and data processing, and data gathering node.

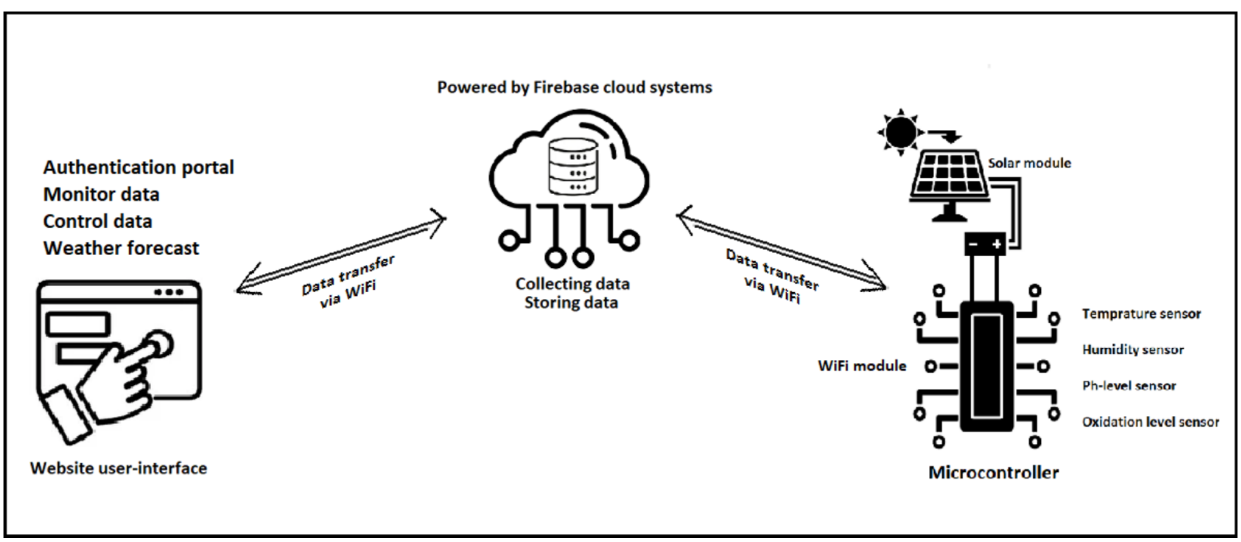


Fig. 3: Basic block diagram

This project will have several parameters and boundaries that will be measured and taken under consideration for our final deliverables. Some of the physical parameters are voltage and power consumption. Since we are working with a lot of digital PIO sensors, we will be dealing with multiple look-up tables that will converge the input voltage data into its corresponding measurement value like temperature or Ph. And for power consumption, we are implementing solar panels that might power up our entire system. Therefore, the power consumption of our entire system must satisfy every sensor connected to it to work properly. Our networking parameters are the transmission latency of our data transmitted and received, the size of data packages, and bandwidth.

The **Fig. 3: Basic block diagram** is created to visualize how our system communicates and transfers data via wireless communication. This diagram represents three major components of our system: User interface device, cloud, and physical node. The node represents our interconnected system with its sensors, power modules, and memory cards. Cloud will be the middle man in this system that will store, and sort data before transmitting it. The user interface is going to be an application and website that will be implemented by our team members to monitor, control, and modify data collected from our system and return feedback as user input. This is just a basic block diagram that represents our system in a simple way to visualize. Detailed block diagrams could be found under their own specific sections named their own components.

A. Physical project breakdown

1) Choice of microcontroller: After conducting some research and laying down all the factors and features provided, we concluded to complete this project with Arduino microcontroller. The model we are going to work with is Arduino Uno Rev2 WiFi, ESP32, and ESP8266 Wrover D1 R1 [9].

Some of the reasons why we made this decision are listed:

i) The prices are the most affordable in the market when it is compared to its performance among other microcontrollers like Rasberry Pi, Edison boards, and Texas Instruments. Arduino Uno Rev2 only costs $ 53.80 that comes with features such as WiFi communication port, up to 14 PWM I/O ports, 6 serial ports, 6 Analog ports, a USB port, and many others. ESP8266 Wrover D1 R1 only costs $ 4.99, and ESP32 only costs $ 11.99.

ii) It is easy to replace and has a fast delivery time since we can access the microcontroller from its own USA manufacturing store as well as Amazon or online sellers.

iii) The components and sensors connected to it are also affordable and cheap since the microcontroller is the most popular among the others. This also allows us to be more flexible with the sensor manufacturer for our own specifics as well as flexibility on replacement parts in case we burn something while implementing.

iv) There are tons of resources available to further expand our knowledge and understanding of Arduino boards and their communication protocols which will be helpful throughout the project.

2) The stingray module: First, Our system will contain a total of three sensors, which could differ for every animal group, a camera module, and an actuator for adjustments in parameters. All of the monitoring module sensors are connected to the I/O pins of the Arduino Uno Rev2 microcontroller. Camera module is connected to ESP32 microcontroller, and Actuator module is connected to ESP8266 Wrover D1 R1 microcontroller. As we can observe on **Fig. 4**.

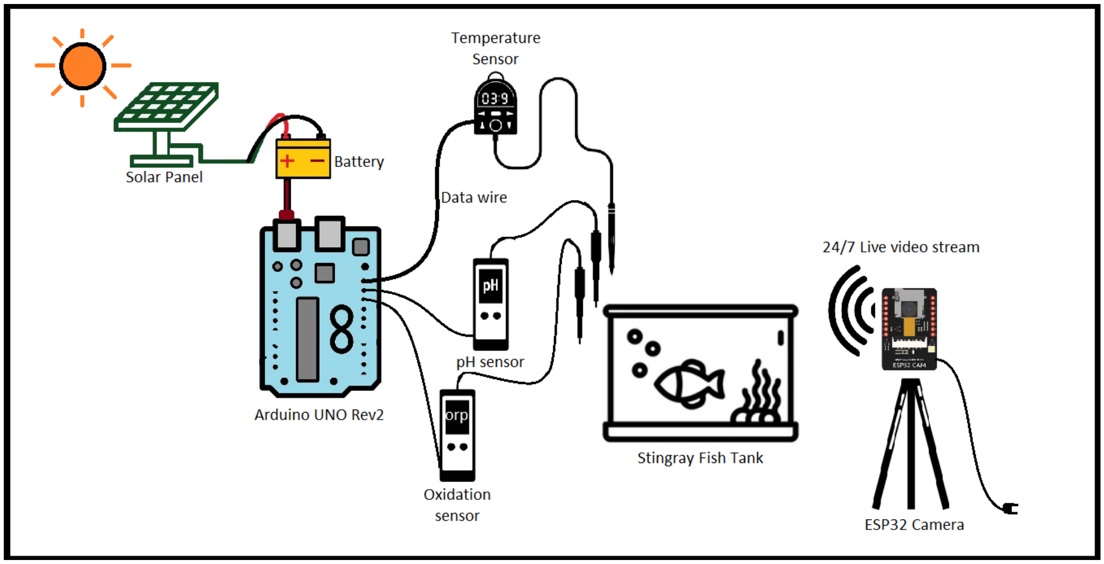


Fig. 4: Stingray module physical diagram

The **Fig. 4: Stingray module physical diagram** is created to visualize how our physical components are standing along and within the complete system of sensors and modules. We have a total of 5 input sensors that will be individually configured and connected to our microcontroller. These are the temperature sensor, ph-sensor, oxidation sensor, camera module, and actuator that will be the controlling part. The data collected will be stored in the SD card module and transmitted via the WiFi module. Furthermore, we also have a physical diagram for our Actuator module demonstrating its design, which could be seen under **Fig. 5: Actuator physical diagram**.

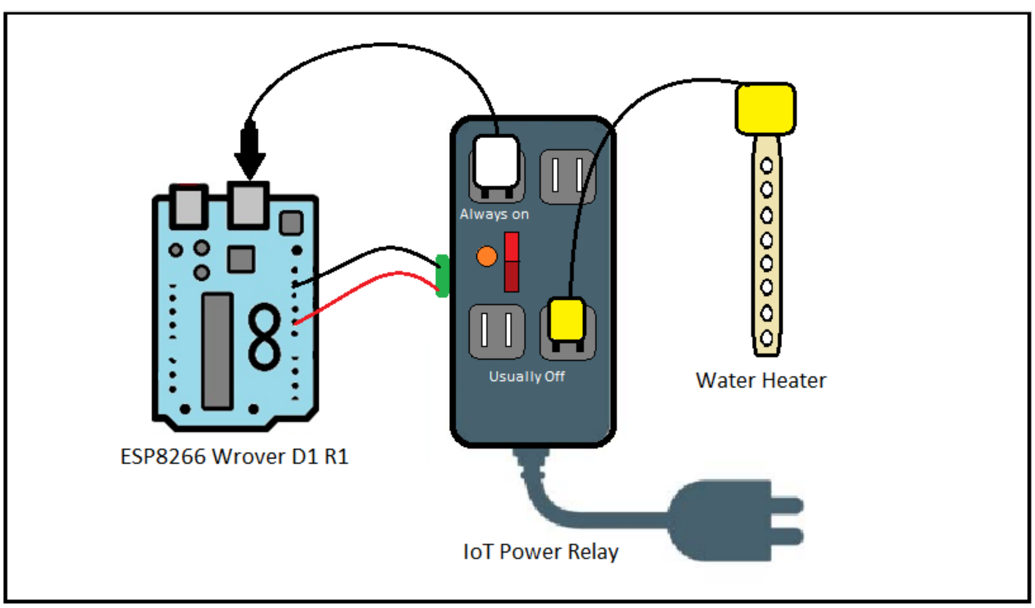


Fig. 5: Actuator physical diagram

Our group has conducted deeper research into sensor connectivity and board pin design, we have completed some basic connections that are required to power up and configure our sensors as shown in **Fig. 6**. There are three sensors connected to our Arduino microcontroller. These are the Ph sensor, camera module, oxidation sensor, and solar module.

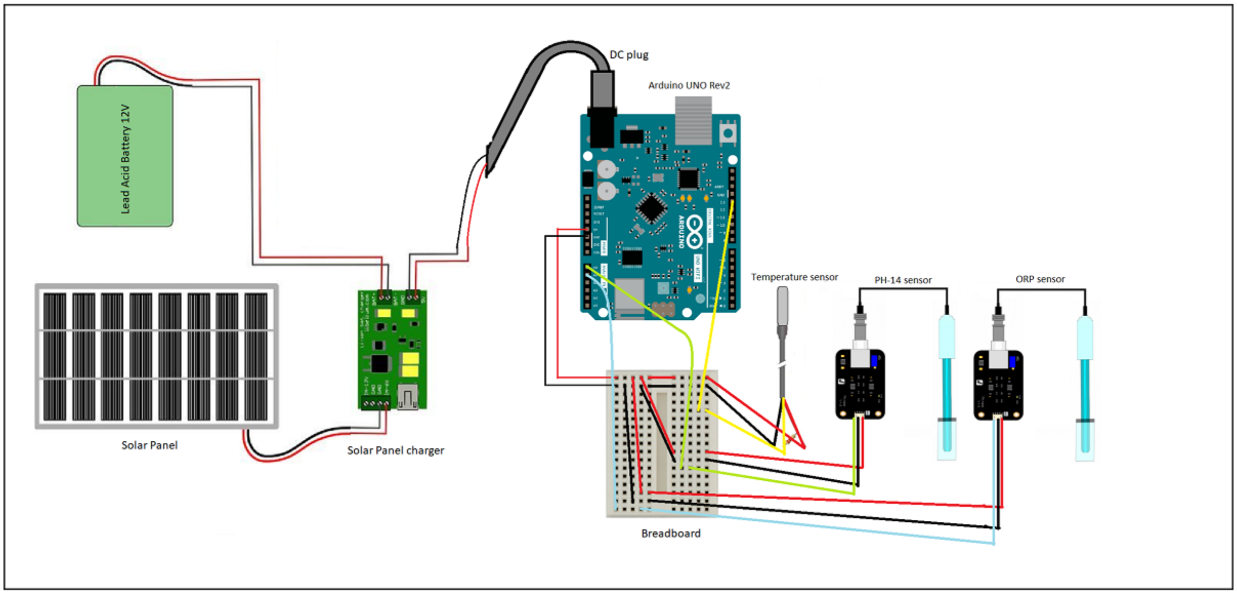


Fig. 6: Arduino and sensors

The **Fig. 6: Arduino and sensors** are created to visualize how our system will look when certain sensors and modules are connected to their corresponding pins. We can see that the figure is demonstrating a total of 3 sensor connectivity including a temperature sensor, oxidation module, and ph-sensor probe. These are all connected to a standard 5V power source and GND. Additionally, they have their own specific data pins that require a connection for them to work. Furthermore, we have a solar module connected as well. The solar module includes solar panels and a battery to charge.

3) The reptile module: Our reptile module consists of a single ESP8266 Wrove D1 R1 microcontroller, one temperature sensor, and one humidity sensor. As mentioned before, under background study conducted about Zoo animals, reptiles require a humid environment to sustain their livings. Therefore, our main focus with this module was to constantly monitor temperature and humidity data. The sensors are connected to the 3.3V output port of the microcontroller, and microcontroller is connected to a wall plug power supply via a Micro USB cable and an adapter. The module is not solar powered since there won't be any sun when this module is located inside the reptile house.

4) The elephant module: Our elephant module consists of a single ESP8266 Wrove D1 R1 microcontroller, one temperature sensor, and one pH-14 sensor. As mentioned before, under background study conducted about Zoo animals, elephant water park require a clean environment to sustain their livings. Therefore, our main focus with this module was to constantly monitor temperature and pH-14 data. The sensors are connected to the 3.3V output port of the microcontroller, and microcontroller is connected to a wall plug power supply via a Micro USB cable and an adapter.

5) Major components: Our Major Component list and their specifications are provided below:

i) **Temperature sensor:** Temperature of the Stingray water is one of the important factors when it comes to providing a sustainable environment for stingrays. The temperature sensor is available for all three microcontroller brands at affordable prices. For the Arduino microcontroller, we decided to use Gikfun DS18B20 Temperature Sensor[10]. This sensor has features compatible with areas in thermostatic controls, industrial systems, consumer products, thermometers, and thermally the sensitive system, etc. This sensor will give us accurate the temperature reading on demand and allow us to monitor the water 24/7.

Power supply range	3.0v to 5.5v
Length	Approx.100 cm
Type	unique single bus interface

TABLE I: Temperature sensor specifications

ii) **Ph sensor:** As this component is one of our major data input sensors, we researched and found GAOHOU PH0-14 Value Detect Sensor Module + PH Electrode Probe BNC [11] that is compatible with Arduino boards and is cheaper than other Ph sensors found while researching online compare to the ones that are compatible for other microcontroller boards.

Detectable concentration range	3.0v to 5.5v
Detection temperature range	Approx.100 cm
Response time	unique single bus interface
Component power	0.5W
Output	analog voltage signal output
Module size	42mm × 32mm × 20mm.

TABLE II: PH sensor specifications

iii) **Camera module:** This component will allow us to detect motion during night time as well as general monitoring purposes when it comes to viewing the stingrays if they are sleeping or moving around. The camera module is available for all microcontrollers listed in this document and they are equally the same price. However, since our group has decided to work with ESP32 camera board, we will be separating it from our main microcontroller, Arduino Uno. This way we will make sure to power our ESP32 with a wall plug to provide maximum efficiency with minimum delay.

ESP32 camera module	OV2640
Build-in 650nm IR block filter	
M12 mount or CS mount lens holder	with changeable lens options
I2C interface	the sensor configuration
SPI interface	camera commands and data stream
Power Source Type	Battery powered

TABLE III: Camera module specifications

iv) **Oxidation sensor:** This sensor is a little more expensive than other sensors that are used in this project. After researching each microcontroller's compatible sensors, the price for Arduino oxidation sensors were more affordable than the others. This module is called Gravity: Analog ORP Sensor [12]. This module will allow us to do certain water quality measurements.

Module Power	+5.00V
Module Size	40mmX27mm(1.57"x1.06")
Measuring Range	-2000mV—2000mV
Suitable Temperature	5-70°C
Accuracy	±10mv (25 °C)
Response Time	20sec
ORP Probe with BNC Connector	
PH2.0 Interface(3 foot patch)	
Zero calibration button	
Power Indicator LED	

TABLE IV: Oxidation sensor specifications

v) **WiFi module:** This module will allow us to communicate our data collected and stored in our SD ram. The data collected will be packaged and sent to the cloud [13] with the help of a WiFi module.

vi) **Solar panel:** Solar panel will be added to our Microcontroller power system to generate clean energy for one of our sensor components. In this way, we will also be adding some complexity to Electrical Engineering parameters. The specifications and block diagram of our solar panel's connectivity are shown in **Fig. 6** that is given under pin assignment section above[14].

vii) **Backup plans for solar panel:** Our team has decided that a solar panel alone won't be efficient for producing power. We have to take precautions when the solar panel doesn't work or is not efficient enough. If the solar panel doesn't produce electricity, then there will not be anything to power our device. Therefore, we will be using an off-grid solar power system. For this, a solar panel will be used as a power source. This power source will then be connected to a solar charge controller. The power will flow through this device. Then the device will be connected to a battery and the component we will be powering. The battery will receive the excess power and will be charged by it. The battery will then be used as a backup power source if the solar panel doesn't produce enough power.

6) Minor components: Some minor components are listed below:

i) **General components:** These include breadboard and jumper wires since this is an ECE project. Therefore we are going to purchase a lot of wires, tools, and resistors required for our sensor configurations, and batteries to get everything working. We could use the department's tools and boards, however since we want to keep this project after we finish this year, we wanted to make sure we are the owners of every single part of this project when it's complete. The breadboard we are going to use for our project is a total of 4-piece breadboards kit that includes 2 piece with 830 points and 2 pieces with 400 points solderless breadboards [15] and we have 40-pin jumper wires.

ii) **Firebase Cloud:** For this project, we decided to have our data stored and monitored inside a cloud base storage [13]. We researched available options for us and came up with Firebase cloud storage packages [13] since it also offers some mobile user-interface application integration along with up to 12 months of a free trial.

iii) **Cables for WiFi and USB module:** To be able to establish communication between our microcontroller and the outside world, we need to connect USB A to B wire in between the Arduino board and WiFi module. This is just a single USB A to B cable. Furthermore, for ESP8266 and ESP32 boards we are using a Micro USB cable to power it up.

iv) **Extra storage:** Since we are unsure how much data will be stored in the SDRAM available in our microcontroller, we wanted to take safety precautions and purchase an extra storage [16]. This storage will be more than enough to store all the input data that are coming from our sensors and nodes even after weeks or months. In this way, we can have backup data information stored in our storage as well for later if anyone wants to check the logs.

Data Transfer Rate	500 MB per second
Digital Storage Capacity	120.0 GB
Hardware Interface SATA	3.0 Gb/s

TABLE V: External Storage specifications

v) **Waterproof containers:** Since our stingray module will be located next to the stingray fish pool, we wanted to make sure our container for our microcontroller and sensors is enclosed with a solid water proof container. We purchased the Flemoon Outdoor Electrical Box, which includes IP54 waterproof outdoor extension cord safety cover, with an outdoor outlet and plugs protector.

vi) **Solid steel containers:** For our elephant and reptile containers, we won't be needing a waterproof container. In this case, we purchased two 6x6 solid steel containers to enclose our required equipment for reptile and elephant modules.

B. Designing the cloud

The data collected from these sensors and modules will be stored first in external storage that will be connected to our microcontroller, and then will be sent to the cloud for data analysis and further data transfer to the user-interface application which is shown in **Fig. 7** [17].

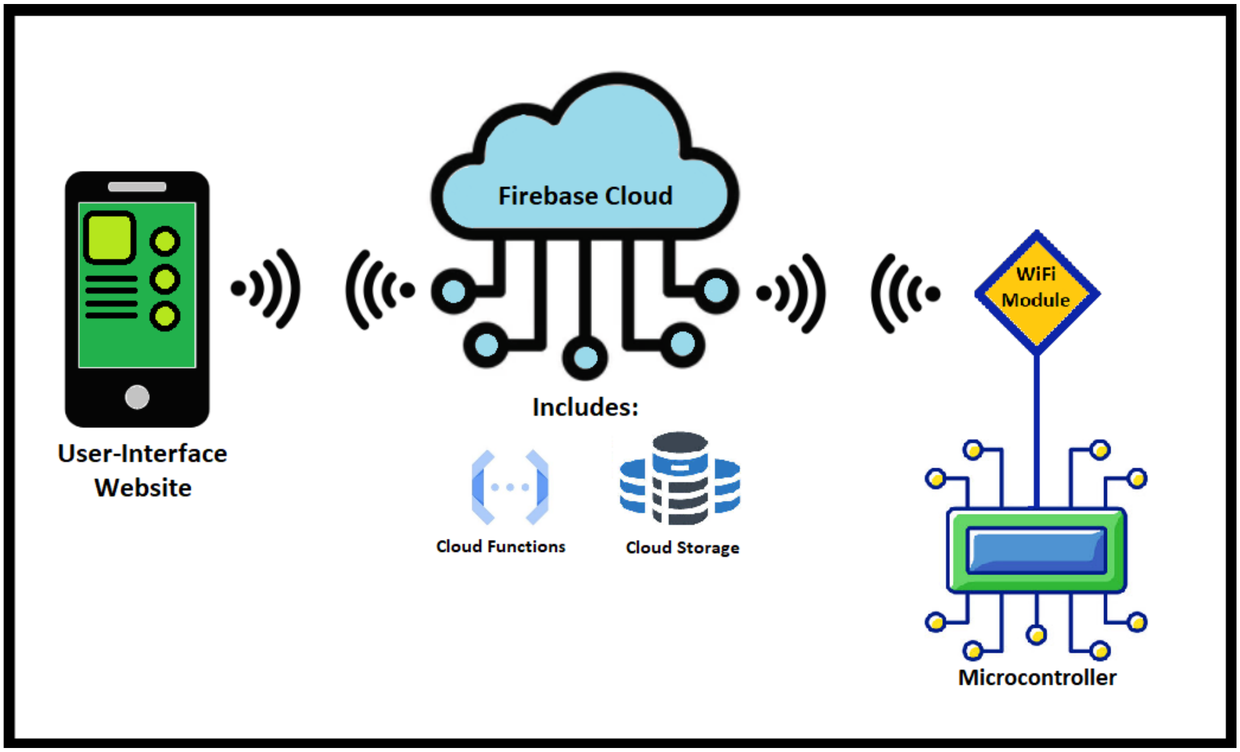


Fig. 7: Cloud block diagram

The **Fig. 7: Cloud block diagram** is provided to help us visualize how the cloud communication system is going to be configured to establish successful data flow between user and node. Our cloud will contain a machine learning algorithm that will provide us with predictions about weather and water temperature to optimize or alert our users regarding any dangers. The cloud is going to store all data collected from our node and sort it in order via the FIFO buffer before transmitting it to our user interface device for monitoring. The user interface will be able to send feedback data which will be stored and transmitted to our node.

As we can see in our block diagram, the data will be transferred and communicated within the node and user input and stored in the cloud itself. With Firebase cloud-based programming we will allow bigger and more acceptable storage for the data transferred from our nodes. Cloud will also be responsible to hold the user input as well which will be used in the control phase of the programming and adjusting the parameters of temperature. Cloud computing will be configured to predict future sensor data, this way, zookeepers can see if there may be any future problems with the environment (ex: temperature lowering over the night) [17].

Detailed explanation of our cloud functions: functions handle all of the requests/receive data from the sensors. After data is inserted into the database, the cloud functions handle how all the data is stored and shifted inside the database. The main benefit cloud functions provided us was it allows for faster handling of the data which is orders of magnitude better than just having the Arduino handle the data in the database.

C. Designing user interface

Third and finally, the collected and analyzed data from the cloud will be communicated and sent to our users via WiFi and they will be allowed to monitor, adjust parameters and view the motion of the animal through the camera module with a developed website user-interface as we can observe in **Fig. 8** [18].

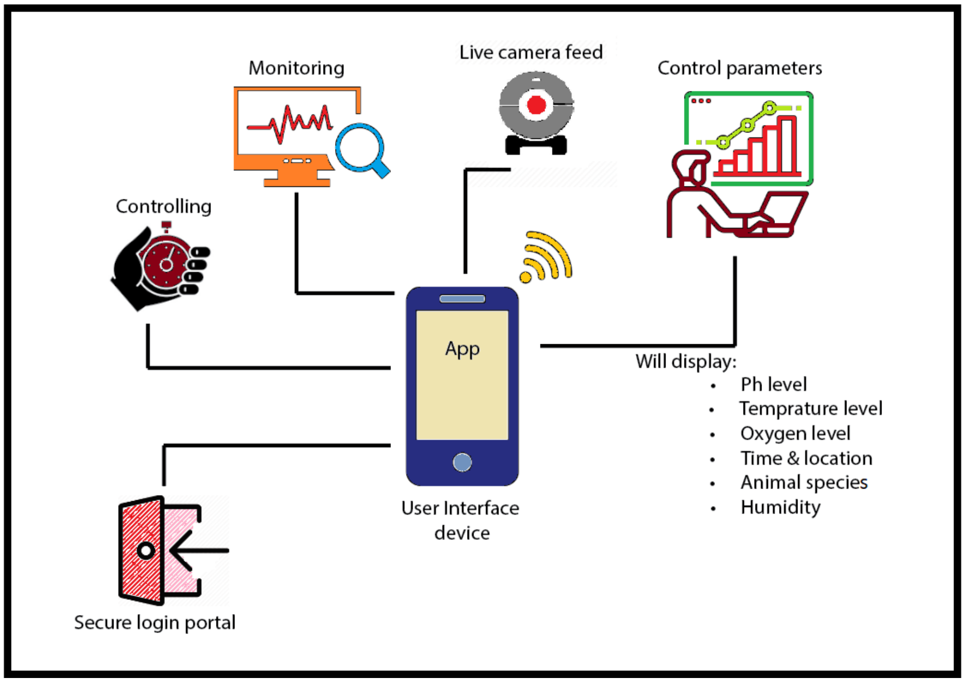


Fig. 8: User interface

The **Fig. 8: User interface** block diagram is created to help us visualize specific components of our user interface module. Some of these modules are the Secure login portal, controlling, monitoring, live camera feed, and control parameters hub. Some of the specifics for the User interface are listed below:

I) **Login portal:** User will log in using the login portal into a specific account to keep track of who made changes.
II) **Cloud:** Data from sensors comes from the database in the cloud.
III) **Display:** We will display the current data and also allow the user to see some of the past data in the form of a graph.
IV) **Camera and motion sensor:** We will allow the user to see the animal through the app using the camera module which will track the animal.
V) **Control Parameters:** We will keep track if any of the parameters reach a critical level, such as temperature which can be adjusted by a thermostat controlled by our microcontroller.
   i) **Ph parameter:** We will show the user how much of a pH controller (Ph up/ Ph down products) should be added to either increase or decrease pH by a certain amount.
   ii) **Dissolved oxygen:** We will increase it using aeration and maintain whatever level we want by controlling how much aeration occurs. Aeration is the process of water coming into contact with air which increases dissolved oxygen levels. This can be as simple as mixing and stirring water [19] so more water comes into contact with air. We could possibly use a small paddle wheel controlled by our microcontroller to control how much aeration occurs.
   iii) **Humidity control:** On soil we could use humidifiers.

VI. High-Level Risks

Some high-level risks contain getting electro-shocked since we are working with water tanks and a humid environment. We need to be really careful with our workstations and proceed to follow certain protocols while handling o while moving around our water tank for test purposes. Other risks include cloud-based programming and user-interface communication between them since our group has little to no knowledge of these topics, we are facing some challenging upcoming tasks that must be completed for our project to be functioning. Machine learning is going to be another challenge for our group to complete, we want to make sure we are as precise as possible with our weather and water temperature prediction system to make the most accurate decisions. And finally, another high-level risk we are facing is not being able to get correct results and outcomes as desired since we are not allowed to test our product on Fresno Chaffee zoo animals due to their high-level security. Therefore, we are going to create a prototype fish tank and create a similar environment and test it that way.

VII. Milestone Schedule Physical

Tasks that need to be completed on a physical level could be divided into sections and subsections to make it visualize better.

A. Configure sensors

This section will deal with the configuration of each sensor that is going to be connected to our node, the microcontroller. Each sensor will have its own data sheet and how to configure it, so this process should not take a long time to be completed. The sensors that will be configured will be the Temperature sensor, pH sensor, Oxidation sensor, Humidity sensor, and the power sensor. Some sensors, such as humidity, will only be configured for areas such as the Reptile exhibit, as they are the ones that need to be recorded for that environment.

B. Configure camera module

Configuration of the camera module is set to be a separate subsection since it is more complex when it comes to configuring the AI motion detector and data collection.

C. Configure solar panel

Configuring solar panel units comes with the knowledge that an electrical engineer requires. Luckily we have a team member that is going to take care of this part and configure the connectivity of our solar panel into our microcontroller and make sure it's functioning.

D. Configure WiFi module

Even though the WiFi module will be available on our Arduino microcontroller, we still need to configure its connectivity to the internet and make sure the data flow is successful and working well.

E. Configure external data storage

This is the part where we connect our external storage to our microcontroller and assign it as our main memory to handle all the storage that will be used later on for data transfer and relevance.

Puya	Responsible for tasks A, E
Sahildeep	Responsible for tasks D
Dominic	Responsible for tasks B
Rafael	Responsible for task C

TABLE VI: Tasks assigned

VIII. Milestone Schedule Cloud

Tasks that need to be completed on a cloud level could be divided into sections and subsections to make it visualize better.

A. Register

We first need to register a Firebase cloud package that is proposed in the paper and further register our networking parameters and configure our WiFi module in our physical node to establish successful communication.

B. Cloud-Node

Establishing successful communication between the cloud and node requires a deep understanding of cloud-based programming, data storage, structure, and stacking. Therefore, there will be a better understanding of these topics listed in the following weeks, and get started with this part once we are done with the physical configuration so we can work with some input data that will be stored.

C. Data analyze

The data collected will be analyzed and programmed here in a way that it will register and give feedback to our node and user if there are any critical levels within the parameters.

D. User-Cloud

Establishing successful communication between the user interface first requires having the user interface ready and implemented. Therefore, this task is unique by having its own prerequisite that will be classified in our Gantt chart. Once a user interface is established, then this task will take place to work on data communication in both directions to either monitor the data on the user end or send the user input data to our microcontroller.

Puya	Responsible for tasks A, B
Sahildeep	Responsible for tasks D, C
Dominic	Responsible for tasks B, C
Rafael	Responsible for tasks B, D

TABLE VII: Tasks assigned

IX. Milestone Schedule User-Interface

Tasks that need to be completed on a user application level could be divided into sections and subsections to make it visualize better.

A. Web application

Having a web application is going to help us with better debugging our program and project if we encounter any problems with monitoring and controlling the data. Therefore, we put this task at the front gate for quality purposes. This task will complete a user-end web application that will have its own web portal to access the website. Once it's accessed, the user can monitor and control the data presented on their end. The data will be communicated via configuration done at cloud-based programming.

X. Milestone Achievements

Our group has completed all of the milestones listed above in all three sections including, physical, cloud, and user-interface parts. There were difficulties and challenges that came up on the way of complication, however, those were successfully over passed and so far it is a completed system that does the following: Our stingray module is capable of collecting accurate temperature, oxidation-level, ph-level, data within a 30 minute time interval and send the collected data to our firebase database. The data could also be collected instantaneously whenever the user desires with a single click of a button. The data for up to two days will be stored in logs of our database for graphing and observation purposes which later will also determine some predictions via a machine learning algorithm. Finally, the user interface has been completed, which now is capable of viewing the data collected and stored in our firebase, adjusting the temperature parameter of the stingray module, and viewing the current temperature of Fresno, CA.

Furthermore, for the reptile and elephant module, their milestones have also been completed, and everything is operating as expected. We are able to collect temperature, humidity, and pH-14 data thru our sensors and store them similarly to our Firebase storage. As our website interface, these modules have their own specific page in our home webpage that users can navigate and monitor the data.

Finally, our camera module has been completed. This was one of the most challenging parts of our project. It has caused so many difficulties due to constraints such as stream delay, connection issues, and lack of http and https flexibility that the website offers for our environment. However, it is currently active and working.

Our complete system's image is provided below in **Fig. 9.**

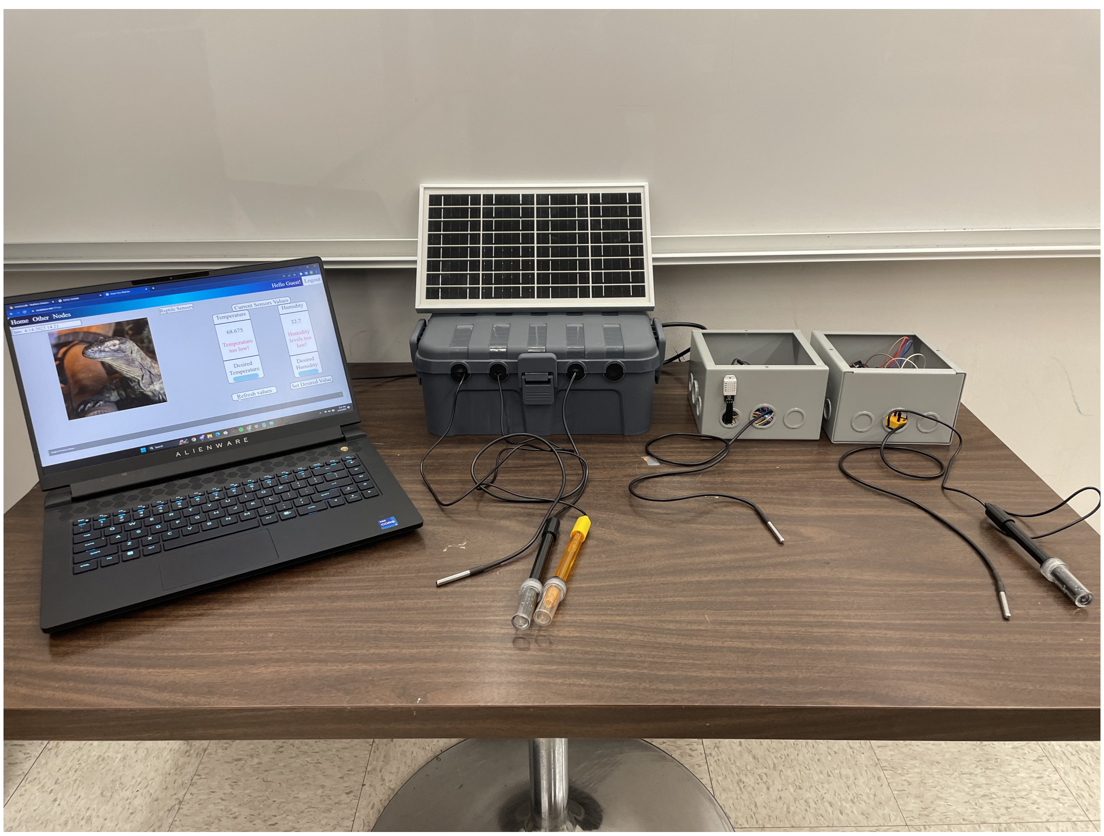


Fig. 9: Complete system

A. *Physical milestone achievements*

The physical components that have got configured and operating successfully are:

I) **The stingray module:** Achievements for stingray module are listed below. It consists of three sensors and a solar module. **Fig. 10.** demonstrates the complete stingray module.

Fig. 10: Stingray module

i) **Temperature sensor:** This sensor was very strait forward when it came to implementation. There was a total of three pins required to be connected to our Arduino board to function: Vcc, Gnd, and DigitalData pin that got connected to pin13 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its temperature level.

ii) **Ph-14 sensor:** There was a total of four pins required to be connected to our Arduino board to function: Vcc, Gnd, Gnd, and AnalogData pin that got connected to Analog0 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its ph level.

iii) **Oxidation sensor:** There was a total of four pins required to be connected to our Arduino board to function: Vcc, Gnd, Gnd, and AnalogData pin that got connected to Analog1 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its oxidation level.

iv) **Solar module:** Our solar module has faced several changes in its module as we faced several difficulties due to the complexity of our system. Our group first wanted to implement the power module via a big acid battery, however, late we changed it to two 9800mah batteries that generate 7.4 V and required a number of amps to power our entire system and sensors. However, that failed too. Now, this system contains a new battery module, EXP1270 12V-7AH/20HR lead acid battery. The microcontroller drains 80mA-100mA when its turned on and fully functional operating. Power analysis has been conducted for this module, which indicates how long our system will last once it's not charged by solar and using the batteries stored. This could be found under its own subsection "Power Analysis."

II) **The reptile module:** Achievements for reptile module are listed below. It consists of two sensors. **Fig. 11.** demonstrates the complete reptile module.

Fig. 11: Reptile module

i) **Temperature sensor:** This sensor was very strait forward when it came to implementation. There was a total of three pins required to be connected to our Arduino board to function: Vcc, Gnd, and DigitalData pin that got connected to pin13 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its temperature level.

ii) **Humidity sensor:** There was a total of three pins required to be connected to our Arduino board to function: Vcc, Gnd, and DigitalData pin that got connected to pin12 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Reptile house to measure their humidity level.

III) **The elephant module:** Achievements for elephant module are listed below. It consists of two sensors. **Fig. 12.** demonstrates the complete elephant module.

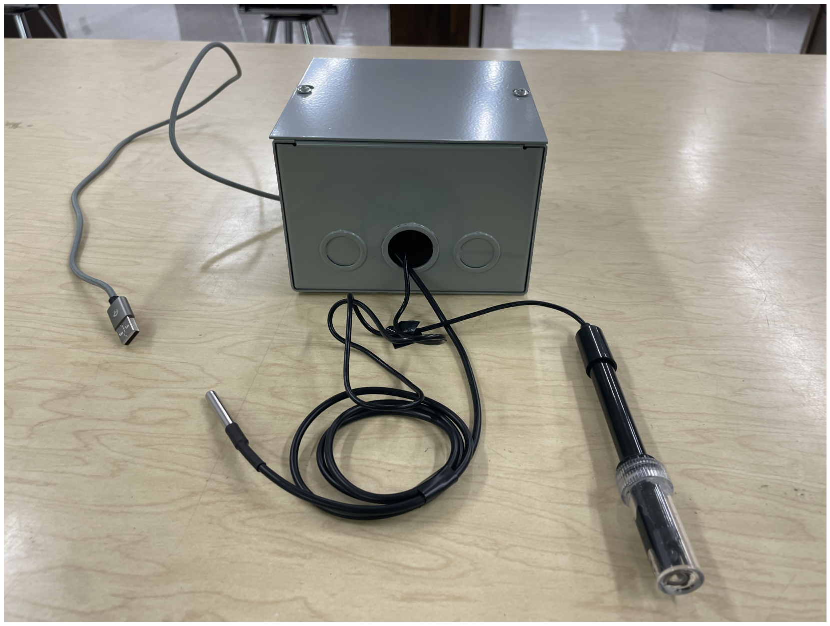


Fig. 12: Elephant module

i) **Temperature sensor:** This sensor was very strait forward when it came to implementation. There was a total of three pins required to be connected to our Arduino board to function: Vcc, Gnd, and DigitalData pin that got connected to pin13 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its temperature level.

ii) **Ph-14 sensor:** There was a total of four pins required to be connected to our Arduino board to function: Vcc, Gnd, Gnd, and AnalogData pin that got connected to Analog0 of our microcontroller. After, our group configured its required software inside Arduino IDE which made it function properly within 30-minute intervals and send data to our firebase cloud to its corresponding variable address. This sensor will be used for the Stingray tank to measure its ph level.

IV) **Actuator module:** The actuator module was one of our most difficult milestones in this project. Our group has went back and forth on deciding what kind of an actuator to implement for the project. This was due to inconsistency at our power module because we first wanted to use Arduino's power output ports to power our actuator, but then it was understood that it is impossible. Then, we finalized our module with the idea of smart IoT power relay module. Power relay module is controllable via microcontroller data input. We have implemented a design where the relay turns on and off using WiFi communication through cloud functions.

i) **Setup:** Setting up the actuator module was very strait forward. First, it required setting up correct configurations for the microcontroller that is connected to it. The power relay contains a controllable data and ground pin on its side, where the microcontroller is able to turn on and off two of its switches, where there are four total. One of the switches are always on, and this feature of the relay came in very handy because we connected the microcontroller to it and made sure microcontroller is always on and ready to turn on the rest of the switches whenever its told to. The data pin from microcontroller is configured to be set on or off via cloud communication, and that is connected to our currently deployed website.

ii) **Current use:** Currently there are two water heaters connected to the power relay, and a microcontroller that will control it. The relay is turned off and on via user input on website. The user will set a desired temperature on their website screen, and the actuator will turn on by reading the on signal coming from the cloud. The main stingray module will constantly check the temperature of the water, and once it reaches to the desired temperature, it will command the microcontroller connected to the relay to shut off. Our power relay module is enclosed inside a waterproof container and could be visualized under **Fig. 13.**

Fig. 13: Actuator module

B. Power consumption analysis

Our group has conducted a power consumption analysis for our system to predict how long the battery used by our solar module is going to last.

I) **Basic understanding:** Our power system consists of three main components which are a solar panel, a sealed lead acid battery, and a solar charge module. The solar panel inputs voltage/current to the solar charge module, and the solar charge module then outputs a fixed voltage/current to the battery to charge it. This fixed voltage/current can be adjusted with a potentiometer attached to the solar charge module. We then connected the solar charge module output and the system in parallel with the battery.

II) **Details:** The voltage from the solar panel will have to be adjusted by the potentiometer due to the change in sunlight throughout the day. The amount of sunlight will determine whether the solar panel or battery will power the load. If the solar panel produces a amount of power significantly higher than what the system needs then the solar panel will both charge the battery and power the system. If the solar panel is producing less power than what the load requires, then the battery will be powering the system. The potentiometer acts as a voltage divider, and we will aim to adjust it so that the solar charge module produces 13.5V-13.8V output since this is the recommended charging voltage of our battery.

   i) **Components:**
      a) 12V 7AH Sealed Lead Acid Battery
      b) 12V 10W Solar Panel
      c) 12V Solar charge module with an adjustable potentiometer and maximum 1A charging current

   ii) **Total Power needed to power our system:**
      a) Voltage: 5-20V.
      b) Current: We measured current to be 50-100mA, but we went with 100mA just to be safe.
      c) Power: We use the power formula to find the total power our system consumes. P = V X I = 12V X 100mA = 1.2W.

   iii) **Power produced by our power system:** Calculations are based on ideal sunny conditions.
      a) Voltage: 12V
      b) Current:100mA
      c) Power:V x I = 12V x 100mA = 1.2W

   iv) **Power generated by solar panel:** The power generated by the solar panel varies due to the amount of sunlight. The solar panel we are using produces 10W and 12V in ideal sunny conditions. This results in a current output of 0.83A for the solar panel. Weather conditions are not typically ideal, so we will rely on the solar charge module to adjust the voltage/current.

   v) **Power generated by solar charge module:** The solar charge module has a maximum charging current of 1A. This means that even if its output current exceeds 1A, the solar charge module will only allow 1A output. We can adjust the potentiometer at a resistance that will give us the exact current and voltage that the solar panel produces. If the solar panel is producing more than 12V (More than what is ideal) then we can adjust the potentiometer's resistance to get a lower voltage to meet our requirements.

vi) **Expected battery life (without solar panel):** 7AH / (100mA x 0.95) = 73.7 hours

vii) **Battery charging time (time needed to charge dead batteries):** 7AH / 730mA = 9.59 hours.

viii) **Battery charging speed:** Assuming ideal sunny conditions, our battery will be receiving more charge than it is producing to the load. The battery will receive 830mA, but the load will receive 100mA at the same time. This means realistically our battery is being charged at 730mA. If we assume that we have 5 hours of ideal sunlight, then we will have 19 hours where the battery is not being charged. This means that after those 19 hours our battery will be at 5.1AH, 7AH - (100mA x 19H) = 5.1AH. Then if we assume 5 hours of ideal sunny conditions, the battery will be fully charged, 5.1AH + (730mA x 5H) = 8.75AH = 7AH. Anything that is 7AH or over is fully charged.

ix) **Nonideal conditions:** If there is less sunlight than the ideal, the system will be powered by the battery. If there is more sunlight than ideal, the potentiometer should be adjusted so that the output voltage of the solar charge module is what we want it to be. All calculations made are based on ideal weather conditions, but since weather isn't always the same then this means parameters are always changing thus the current that our system receives is changing as well.

x) **Special cases:** The 12V 7AH sealed lead acid battery that we are using has a recommended charging voltage of 13.5V-13.8V. This means that if we have weather conditions that exceed ideal weather conditions for the solar panel, we can adjust the voltage on the solar charge module to produce an output of 13.5V-13.8V. Our system is recommended to take in between 7-12V but it can handle 5-20V. Since the amount of sunlight determines whether the solar panel or battery powers our system, we do not have to worry about our system possibly overheating when receiving 13.5V-13.8V as we charge the battery with this voltage. This means we can use a 13.5V-13.8V output from the solar charge module instead of 12V.

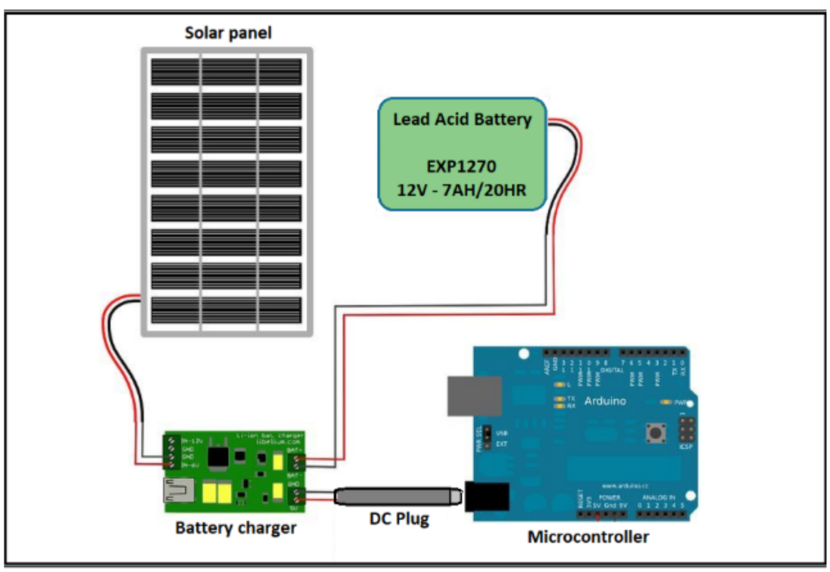


Fig. 14: Solar Module

Fig. 134 given above, is the prototype of the solar module system that we are currently using. This image will be replaced via its actual components once everything is fully connected along with the sensors.

C. Cloud milestone achievements

i) **Firebase cloud overview:** Our group has decided to change the AWS database cloud system to the Firebase cloud system instead. This decision was made to make cloud programming easier since AWS requires SQL database coding language to work with, whereas Firebase doesn't and is more object-oriented. We named our cloud "smartzoodb" and so far have been using it to analyze data collected from our microcontroller and sensor and store the data. Data collected will be stored in our cloud up until the last two days, however, after two days it will be deleted to utilize cloud storage. These are collecting data from our temperature, Humidity, Ph-14, and Oxidation sensor. We are also displaying the current time and date.

Our Firebase Real-Time Database has the structure shown in the figure below.

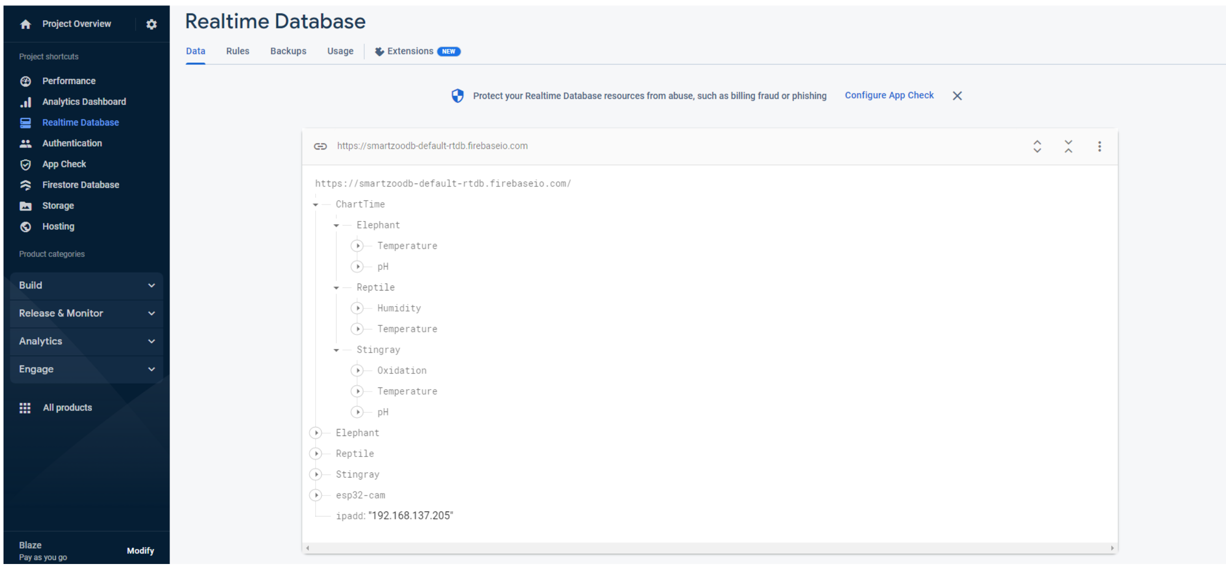


Fig. 15: Chartime data

ii) **Firebase realtime database:** In the figure above **Fig. 15.** we can see the node called "ChartTime". This node stores sensor readings corresponding to the time they were taken. We have 30-minute intervals between collecting data so it is stored in 30-minute intervals at points such as: "1330","1400","1430"... and so on. This data was ordered like this in the ChartTime node so it can be easily pulled from the website, so we can show that data in a chart which gives users another way to interpret the data collected by our sensors.

We can see under the ChartTime node that we have the Elephant, Reptile, and Stingray nodes. These nodes also contain sensor readings for the respective animal and it contains the 48 previous values of the sensors. The main reason we need this and didn't include it in ChartTime node is the user can refresh and request data so we wanted to keep that data but also didn't want to disrupt our 30-minute interval system in the ChartTime node. All the data that is being collected has its path to the sensors in these 3 paths. For example, if the stingray module was updating the temperature it would insert the new data at the path "Stingray/Temperature/Value1" where value 1 holds the latest temperature. Then the cloud functions take over and shift the data so "Value3" becomes "Value2", "Value4" becomes "Value3" and so on till "Value48" becomes "Value47".

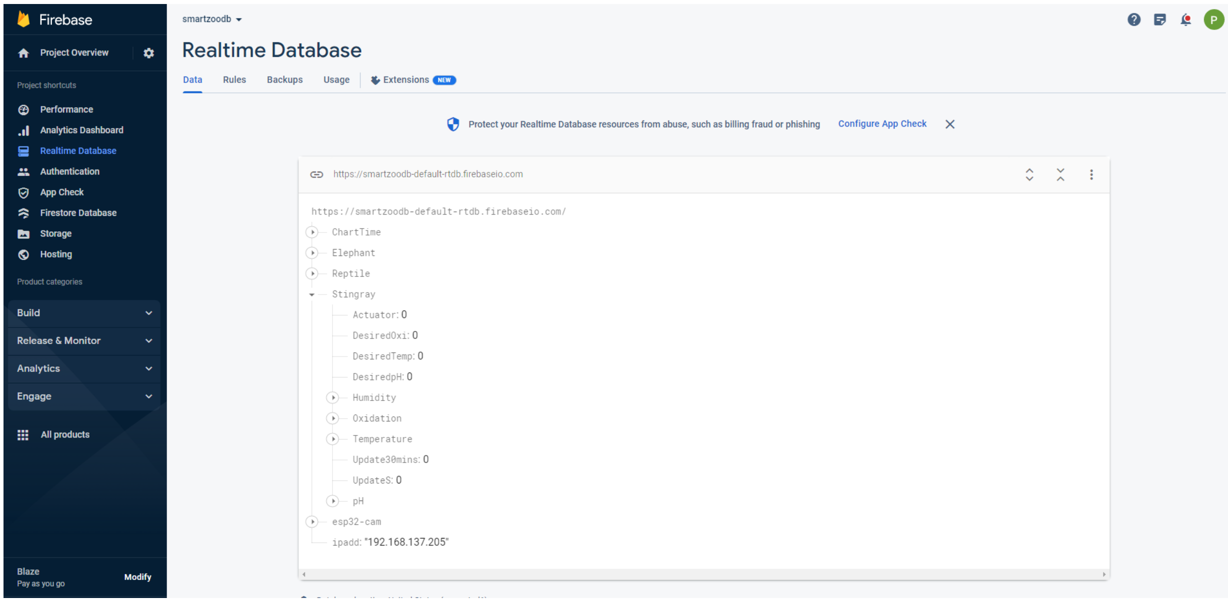


Fig. 16: Firebase data

In the figure above **Fig. 16.**, we have opened the stingray node and we can see all of its child nodes. The “Actuator”, “DesiredOxi”, “DesiredTemp”, and “DesiredpH” are related to the actuator part of the project. The “Humidity”, “Oxidation”,”Temperature”, and “pH” nodes contain the past 48 measurements of the respective sensor. The “Update30mins” is set to 1 every 30 minutes by the “Every30mins” cloud function. The arduino reads this and sends the latest measurements and then the arduino sets this back to 0 again. The “UpdateS” node is connected to the refresh button on the website and pressing that button will set it to 1. The arduino will read this and send the current data into the database and it will set the “UpdateS” to 0. The rest of the animal nodes are similar to this as well. Finally we have the “esp32-cam” and “ipadd” nodes which hold information about our camera module. Our camera’s IP address is always changing so we have the node “ipadd” to hold the camera’s latest IP address. The “ipadd” node is used by the website which reads the IP address of the camera and uses it to display the camera’s live feed on the website.

iii) **Firebase cloud functions:** The cloud functions were a important part of this project because it took burden off the arduino microcontrollers which provides us with more battery life since it has to compute less things. At first, we had the microcontroller insert and shift data around in the real time database. We measured this and used to take about 3 minutes for the arduino to insert and shift the data correctly in the database. This is mainly due to the fact that it had to send 48 read requests and 48 write requests to the database for each sensor. We then employed cloud functions into our project and saw that the same task now takes 10-15 seconds.

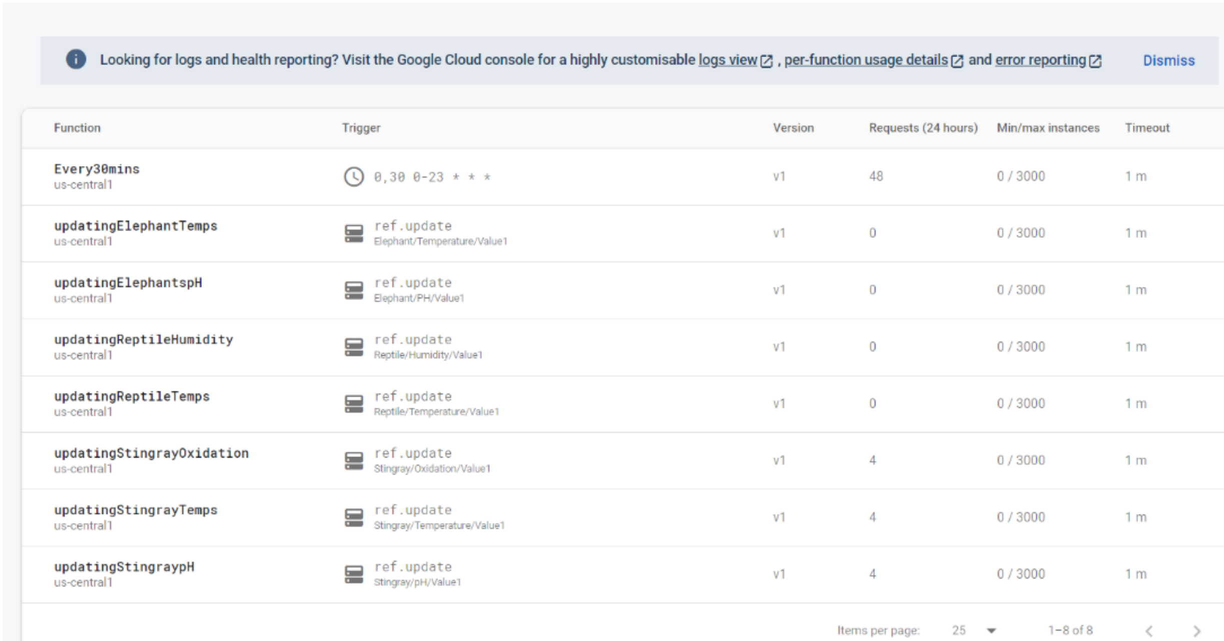


Fig. 17: Cloud functions

Fig. 17. demonstrates the cloud functions, which are written in javascript, and most of the cloud functions are update functions and each sensor for each animal has its own cloud function. These update functions execute when data is inserted into the respective sensors path in the database. Once data is inserted, the cloud function shifts data. The shifting can be thought of as a First in First out (FIFO) data structure where inserting a element will push an element into the front of a queue or list. At the same time, the element at the end of the list or queue is popped and removed which is essentially what this cloud function does. Also this update function also inserts this data into the correct time slot in the "ChartTime" node in the database. This update function only occurs every 30 minutes which is controlled by another cloud function called "Every30mins".

D. User-Interface milestone achievements

i) **Purchased the domain:** The Smartzoo.net domain name has been purchased, and with the help of firebase hosting we are able to host our website at minimal cost for the project. The website currently has a login page for authentication, as well as different sections for each of the nodes we are focusing on.

ii) **User interface login page:** The user authentication page was made using firebase authentication, allowing us to control who views the website and ensure that only authorized users are able to view and change the designated values for what the zoo animal parameters should be. This gives the website added security for these sensitive animals in ensuring that they are not disturbed from their environment by outside malicious entities. Since employees at the Fresno Chaffee Zoo are given their own work email, for this demonstration there was no need for a sign up feature for new users.

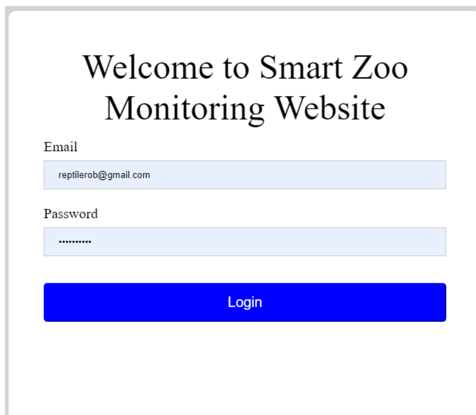


Fig. 18: Log in page

iii) **Homepage:** A homepage was implemented so that upon logging in, users would be able to see the current status for all the different animals at a quick glance. While this does not give an in-depth view of the actual values that are being collected on the sensors, it is a good indicator of how the animals are doing. This homepage also includes a weather forecast section so that users are able to get more details on the weather conditions. This weather is collected from OpenWeather, an API that sends data directly to the website which is broadcasted using javascript.

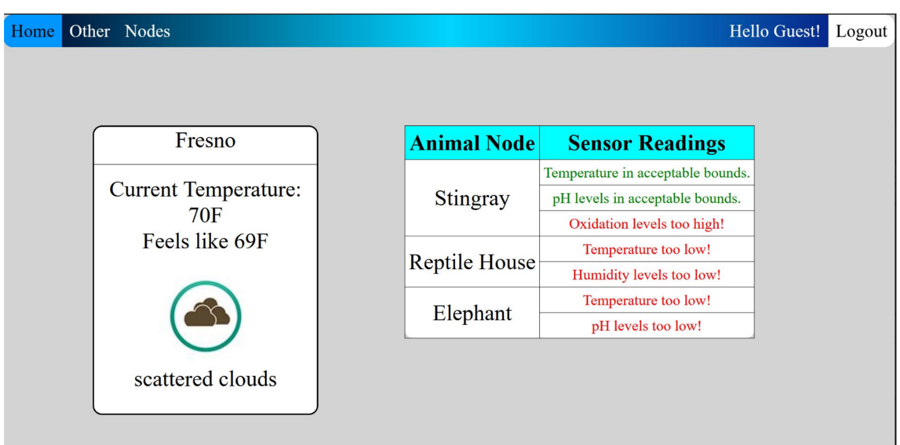


Fig. 19: Homepage

Our current homepage is displayed in the figure above **Fig. 19: Homepage**. As mentioned earlier, the homepage currently includes quick feedback on the parameter checking of our data collection. This is the quickest way to gain information regarding the status of the animal's habitat without going into their own web pages. However, we have more detailed insight into our animal's habitat if the user wants to view the data readings. That could be achieved via going to **Nodes** section from the top navigation bar and clicking the desired environment they want to check.

iv) **Animal's web pages:** With the project focusing on three different animal groups, three different web pages were made that could be accessed from the website's navigation bar. Each of the pages hold the sensors that are specific to their animal, such as stingrays having temperature, pH values, and oxidation. Reptile House nodes will display humidity and temperature, while elephants will have temperature and pH levels. Additionally, the stingray node will have a camera section for an active viewing of the stingrays. All nodes will have a graph showing the previous values over a 24-hour period so that zookeepers can keep track of the trend of the values. All nodes will also have a section to designate desired values which will turn the actuators on to modify the enclosures in real time to match the desired values.

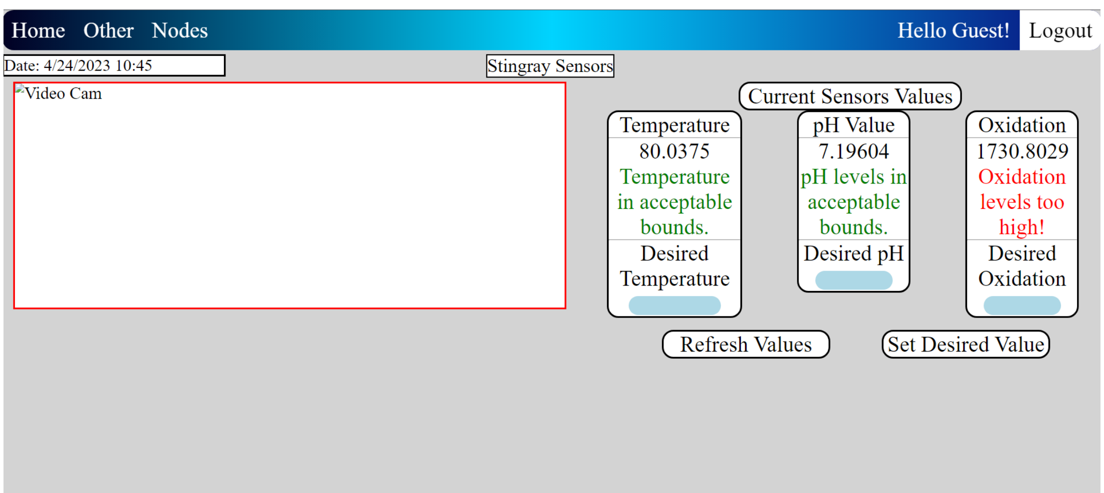


Fig. 20: Current stingray page

The figure above **Fig. 20.** displays our webpage specified for our stingray fish tank. This webpage includes detailed data regarding the collection of data sensors from our microcontroller. We can observe the temperature, ph-14, and oxidation level. Moreover, the data displayed will have a brief notification underneath which purposes parameter check of the animal habitat. In this case, we can observe that the temperature is in an acceptable range, however, ph-14 and oxidation are not. Finally, there will be a live feed of the fish tank via a camera that will be integrated into our Arduino which will detect if the fishes are asleep or not.

XI. Quantitive Analysis

After successfully completing and fully operationalizing our system, we conducted quantitative analysis to measure the delay being experienced within it. This analysis involved collecting and analyzing data on various metrics, such as response times, processing speeds, and error rates. The analysis were conducted under ethernet wifi connectivity, to calculate best results.

I) **Delay from website to cloud:** In order to analyze the delay between the website and the cloud, we must observe the traffic when a user enters a command on the website and watch how long after it will change that variable's value on the firebase. In this case, we have conducted multiple tests and took the average. The delay between the website and the cloud is almost nothing to zero. It cannot be measured in milliseconds.

II) **Delay from cloud to website:** In order to analyze the delay between the cloud and the website, we must observe the traffic when the user make changes inside firebase cloud and time how long it takes to affect on our website. The delay between the cloud and the website is again almost instantanious, cannot be measured with milliseconds.

III) **Delay from cloud to hardware:** When it comes to delaying between cloud to hardware, a lot of factors play an essential role, such as internet speed and hardware wifi module quality, but in our case, we measured it via the user asking the hardware to start the actuator module, this way we can measure the delay sufficiently. The delay in turning on the actuator module from the cloud is about 0.5-1 seconds.

IV) **Delay from hardware to cloud:** In this case, we measured how long it takes for the hardware to send and update new sensor values to our cloud. This is important because this plays the biggest role when it comes to overall delay since the delay between the cloud and the website is very small and could be ignored. The delay between the hardware to the cloud is 1-1.6 seconds. However, there is an important detail to note. Since we are using multiple sensors, the delay for each reading is different. What that means it the delay between the first sensor reading to update on the cloud is 1-1.6 seconds, however the delay for the second sensor reading will take an additional second or 1.6 seconds as well.

V) **Overall Delay:** When it comes to calculating the overall delay, we must understand that our system is a two-way communication system in which the user first commands the hardware to collect data from the website, then the hardware receives this signal and does what it is asked to do. Therefore, to calculate the overall delay we have done some sample tests and measured the average delay of our test scores. The delay is 3.2 seconds. The reason for this is quite simple, our reptile module, the module that had been tested for delay analysis, has a total of two sensors. And we calculated that each sensor reading from cloud to hardware takes 1-1.6 seconds. Therefore, it is quite reasonable for the reptile module to have a delay of 3.2 seconds because the delay is between the cloud to website ignored and the only delay affecting the overall system is between hardware and the cloud.

XII. Gantt Chart

Our current Gantt chart is given below in **Fig. 21.** with its corresponding dates and timeline. The updated gantt chart indicates that we have accomplished all of the milestones through this semester. Green bars are tasks that are completed.

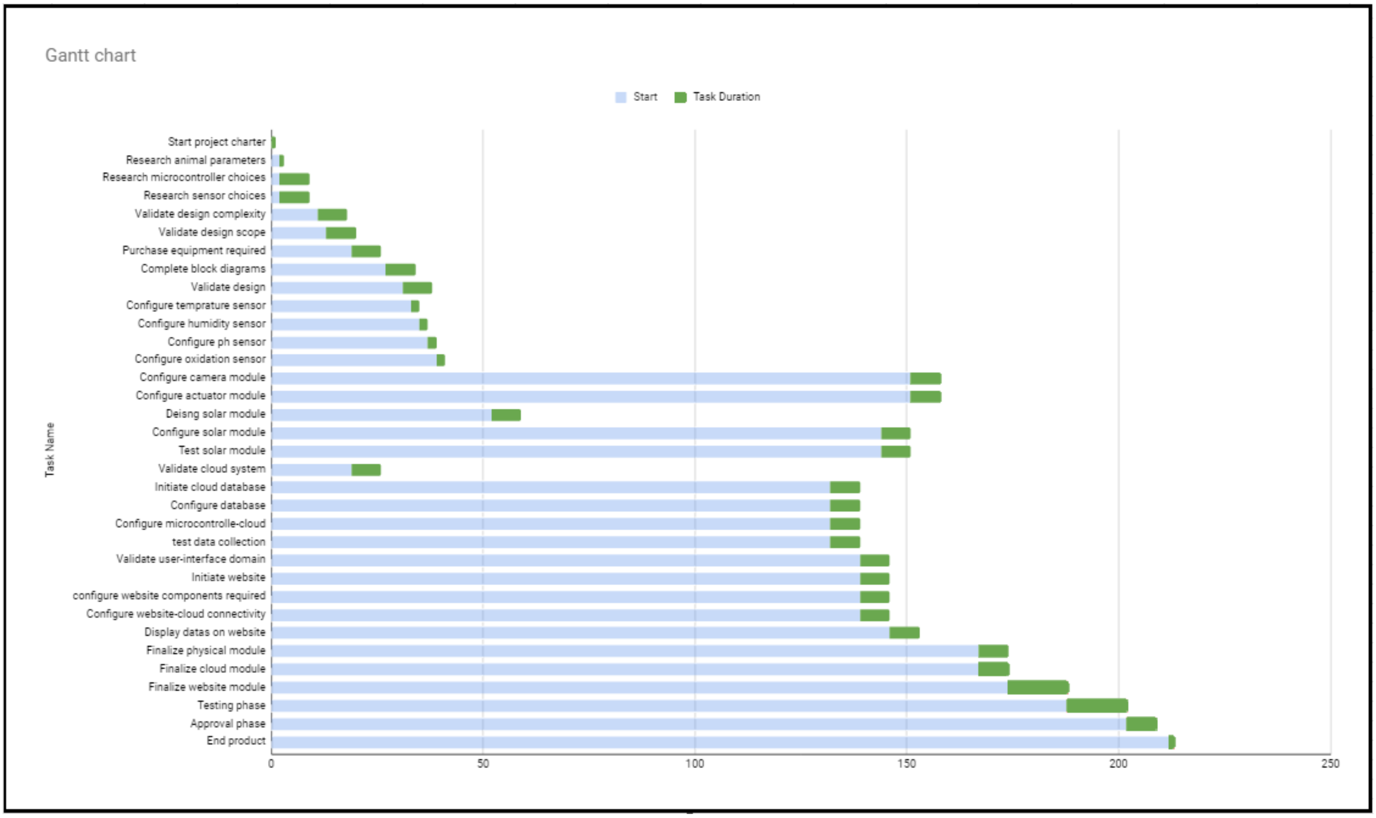


Fig. 21: Gantt chart

Our Gantt chart doesn't specify each individual task for our group members, those are specified under the milestone schedule part of this report. However, this Gantt chart demonstrates detailed tasks that are required to be completed to the achieve end product. As can be observed, we have all completed parts that are shown by green bars.

XIII. Test Plan

Our test plan will have four total major parts and several minor parts that will be needed to be successful throughout our test runs. **Fig. 22.** indicates the big picture of what those major and minor tasks look like.

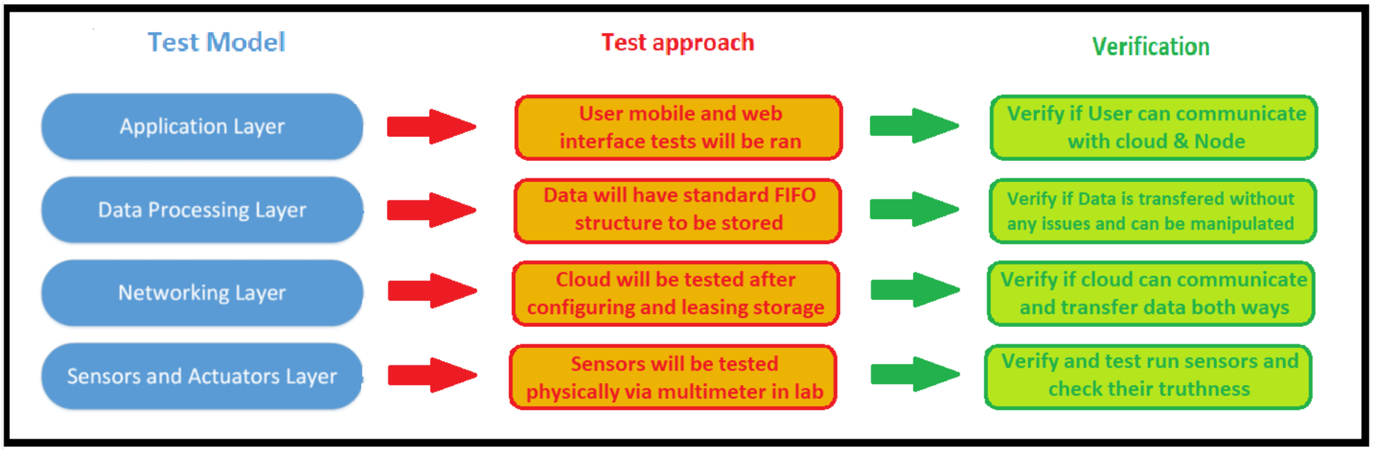


Fig. 22: IoT test model

As shown on **Fig. 22.**, test model is created to help us visualize the basics of the test model and verification protocols that our team is going to follow. This model is generated by using our IoT model from our **Standards** section of this report. Our IoT layers will have their own individual-specific test approach and verification protocol. Specifics of these layers can be found in the description paragraph below.

I) **Plan identifier:** We will split the test plan into four phases. The first phase would be testing the individual input-output response of every sensor and actuator we implement in the bottom layer of the IoT model. The second phase will be testing the connectivity of the Networking Layer. The third phase will be completing the Data processing layer and checking if we can store, manipulate, and monitor the data collected from our sensors and nodes. And finally, the fourth phase will be configuring and checking our Application layer. Once every layer passes successful test runs, we will test the entire system together and complete several example samples collected from zoo water and validate our project.

   Outcome: Every sensor has been individually tested and compared against another testing measurement to confirm their credibility. We have successfully accomplished two-way communication for our system, users are able to communicate to microcontrollers through cloud and website interface. Data has been successfully processed by cloud functions assistance and cloud storage. We have a deployed website that users can monitor and control the individual modules. Therefore, this step is complete.

II) **Application layer test model:** In this layer, we will first test run our user and user interface inside the software and debug if there are any occurring problems and issues inside the software itself. Then for the final phase, we will test its communication when its part of the system by sending and receiving data from our cloud and node.

   Outcome: We have a deployed website that users can monitor and control collected data. All errors have been debugged.

III) **Data processing layer test model:** In this layer, we will first create an algorithm that will queue the transmitted data into a FIFO buffer as our structure. Then for our final phase, we will test this buffer when it's part of the whole system and verify if we can transmit the correct data and be able to manipulate it.

Outcome: This test item is associated with our cloud interface and setup. We have successfully registered our Firebase cloud and it is currently operating with no errors. It is capable of transmitting data both way, from microcontroller to user and from user to microcontroller.

IV) **Networking layer test model:** In this layer, we will deal with a lot of WiFi connectivity as well as cloud-based programming and its applications. Therefore, we want to make sure we first have the correct cloud configurations that will allow us to transmit and store data successfully. And for our final phase, we will include it in our final system and verify a both-way communication.

Outcome: Cloud functions have been implemented to decrease the connectivity delay when the user wants to collect data. This step is also complete.

V) **Sensors and Actuators layer test model:** In this layer, we will be mostly dealing with physical parts of the project's configuration and dealing with their input data. We first want to ensure we get correct readings from our sensors that will align with correct look-up tables for things like temperature and Ph since they are all in voltage readings. And finally, we will verify it when it is part of the whole system and how the input data is transferred to the cloud, and the user application.

Outcome: The actuator module is complete and operating successfully. The group has tested the actuator module separately and it is doing what it is supposed to. Heating up the water, and shutting off when the temperature is reached to the desired value.

VI) **Test items:** All of the items listed under the physical sensor and node configuration, as well as cloud, and user-interface application will be tested by the end of its own Gantt chart deadline.

Outcome: All of the items listed on Gantt chart are complete and tested. Furthermore, there has been a sample test conducted at Fresno Chaffee Zoo at Stingray water park. It has been successful.

VII) **Test pass/fail criteria:** A successful range of data collection is required for our system to work and function. Therefore, it's really critical that the Sensors and Actuators Layer of our model successfully passes all the test runs.

Outcome: The system has passed all the tests.

XIV. Real World Experiment

Fig. 23: Test conducted at zoo

The figure above **Fig. 23.** displays our exhibit visit at Fresno Chaffee Zoo. Our group has brought the entire stingray module to the zoo and ran several test runs by the Stingray water tank. The temperature, pH, and Oxidations level shown at our website were accurate with the ones Zoo measured at the backstage. Furthermore, we were also able to capture a live feed with our ESP32 Camera module.

The test station was placed at the main stingray water park, closer to the back area so that visitors won't be bothered. The total time spent at the zoo was four to five hours, which is enough time to collect a total of 10 sample data for each one of our sensor readings. It was noticed that the temperature and pH sensors calibrate within minutes and give the correct readings. However, the ORP Oxidation sensor took about 30-1 hour to calibrate before collecting the correct and accurate readings from the water tank. This is due to the sensor quality. After all, the sensor being used in this project is budget friendly and not the best in the market.

Our camera module was also enabled during the experiment, and we were able to detect and watch stingrays swimming inside the pool clearly. However, there were times that the camera feed froze because the internet being utilized at that current moment was supported via a mobile hotspot.

XV. Projects Day Presentation

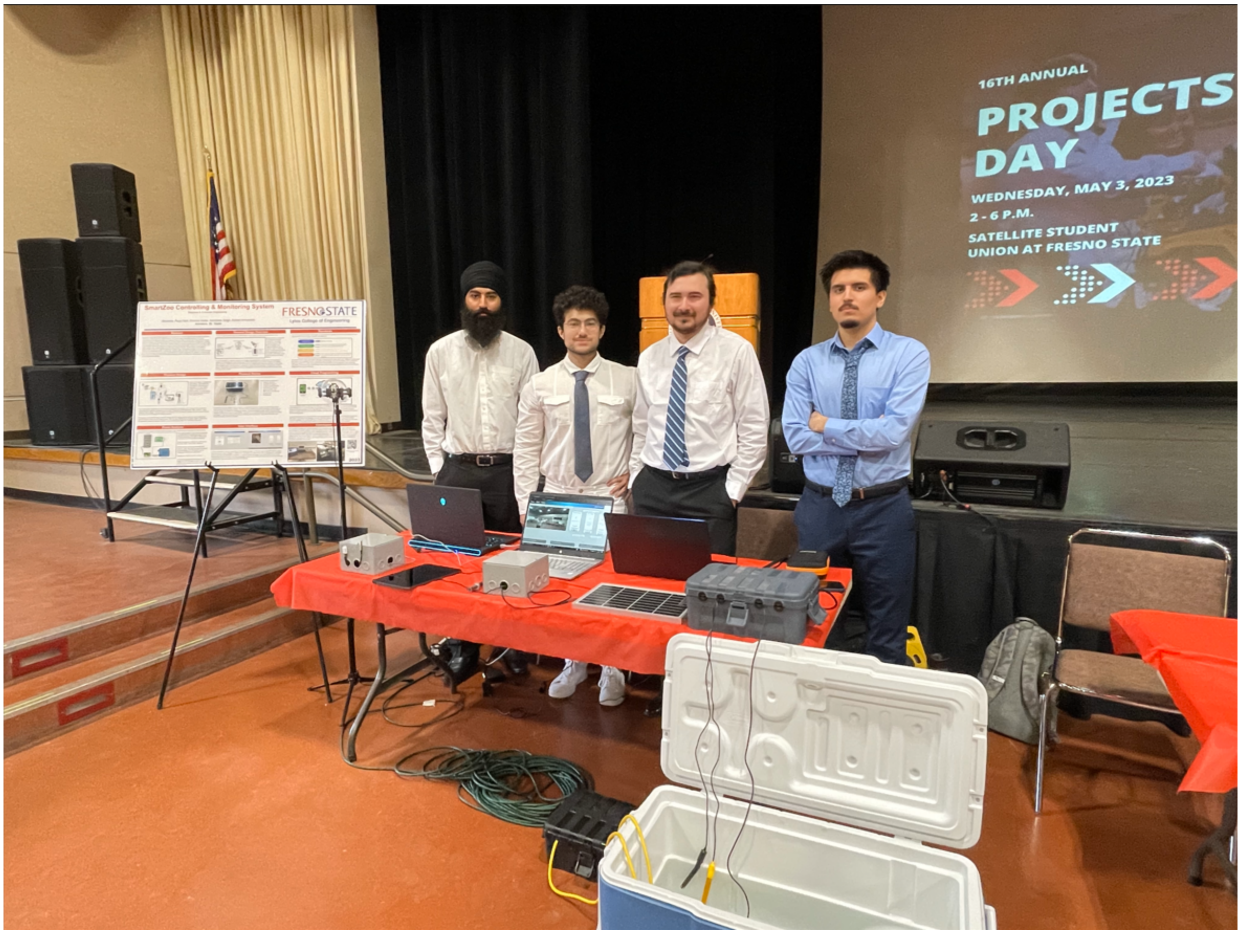


Fig. 24: Projects day

Our group attended the projects day presentation to demonstrate the working project to our faculty and other university visitors. The presentation went smoothly and successfully, with more than 15 visitors who came by and were curious about our project. Each group member was assigned a role beforehand to make sure we are fully prepared. Puya was responsible for introducing the project to anyone who walks by and going over a brief demo and explaining individual components. However, if the question asked got technical in areas of cloud, website, or power, Puya would refer the visitor to Sahil, Dominic, or Rafael in order.

For demonstration, we have set up an ice cooler station filled with water because our stingray module only operates inside water. This way, we have demonstrated our entire stingray module as well as our actuator module, which will heat up the water to a certain degree that is set by the user. We also brought some ice to cool the water back down after heating it up multiple times, to avoid boiling temperature. We have used ethernet connectivity to reach maximum performance with our system and camera feed while providing minimum delay.

Overall, Projects day was successfull and took about 8 hours in total, 4 hours of preperation and 4 hours of demonstration. Afterwards, all the equipment went back to ECE186B lab room.

XVI. Equipment and Budget

All the equipment for this project will be purchased within our own team member's budget. All the items below have been already purchased and are currently in use.

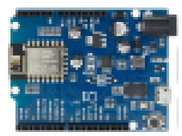
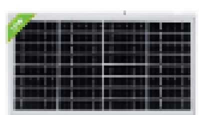
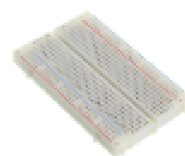
Components	Figures	Components	Figures
Arduino Uno Rev-2		Temperature sensor	
ESP32 Camera module		Oxidation sensor	
ESP8266 Wrove D1 R1		pH-14 sensor	
Solar Panel		Humidity sensor	
Lead acid battery		Breadboard	
Waterproof container		Jumper wires	
Solid steel container		DC Plugs	
USB cables		Extra storage	

Fig. 25: Component table

All the total amount for the purchased equipment listed below:

i) The total budget comes to around $489.00
ii) Breadboard and jumper wires: $10.99
iii) Extra storage: $10.99
iv) Cables for WiFi and USB module: $20.95
v) Arduino Microcontroller: $53.80
vi) ESP32 Microcontroller: $13.80
vii) ESP8266 Microcontroller: $20.80
viii) Temperature sensor: $11.58
ix) Ph-sensor: $28.60
x) Camera module: $25.99
xi) Solar module: $38.90
xii) Oxidation sensor: $ 89.00
xiii) Humidity sensor: $ 5.00
xiv) Waterproof container: $80.00
xv) Solid steel container: $35.00
xvi) Water heater probe: $45.00

XVII. Stakeholder List

List all of the stakeholders for this project.

1. Technical Advisor: Dr. Shahab Tayeb
2. Course Instructor: Dr. Shahab Tayeb
3. Project Manager: Puya Fard
4. Team member: Sahildeep Singh
5. Team member: Dominic Keifer
6. Team member: Rafael Hernandez

XVIII. Project Approval Requirements

This project is required to be a functioning system of sensors and nodes that are all connected within themselves and finally to the internet so that users can monitor and control the system by making corrections and adjustments to these parameters. In conclusion, successful user-end websites applications as well as cloud-based communication for data transfer and collection from our node are required.

XIX. Approvals

This is a list of signatures of all of the stakeholders

Project Manager: Puya Fard ______________________________________
Team Member: Sahildeep Singh ______________________________________
Team Member: Dominic Keifer ______________________________________
Team Member: Rafael Hernandez ______________________________________
Technical Advisor: Dr. Shahab Tayeb ______________________________________

REFERENCES

[1] "Lyles college of engineering." [Online]. Available: https://engineering.fresnostate.edu/ece/tayeb/

[2] "Freshwater stingray care guide." [Online]. Available: https://www.aqueon.com/resources/care-guides/stingray-freshwater

[3] N. Mduduzi and L. Molaba, "Water for african elephants (loxodonta africana): Faecal microbial loads affect use of artificial waterholes," Aug 2018. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsbl.2018.0360

[4] J. Cousins, "Temperature and humidity monitoring for pet reptiles and amphibians," Oct 2021. [Online]. Available: https://www.sensorpush.com/articles/temperature-and-humidity-monitoring-for-pet-reptiles-and-amphibians: :text=The&20most%20important%20thing%20to,reach%20over%20100%C2%B0F

[5] S. J. Lopez and M. C. Louis, "The principles of strengths-based education," *Journal of College and Character*, vol. 10, no. 4, 2009.

[6] "Fresno state." [Online]. Available: https://www.fresnostate.edu/catalog/courses-by-department/elect-computer-engineering/index.html

[7] E. R. Lacy, J. S. Nicholas, and J. Colglazier, "Changes in the composition of coelomic fluid in the atlantic stingray, hypanus sabina, upon dilution of the environmental salinity: osmoregulatory and ph implications," *Ichthyological Research*, pp. 1–8, 2022.

[8] "Iot architecture layers," Jul 2019. [Online]. Available: https://www.hiotron.com/iot-architecture-layers/

[9] "Arduino uno wifi rev2." [Online]. Available: https://store-usa.arduino.cc/products/arduino-uno-wifi-rev2?selectedStore=us

[10] [Online]. Available: https://www.amazon.com/Gikfun-DS18B20-Temperature-Waterproof-EK1083x3/dp/B012C597T0

[11] "Amazon.com: Gaohou ph0-14 value detect sensor module + ph electrode ..." [Online]. Available: https://www.amazon.com/GAOHOU-PH0-14-Detect-Electrode-Arduino/dp/B0799BXMVJ

[12] "Gravity: Analog orp sensor meter for arduino." [Online]. Available: https://www.dfrobot.com/product-1071.html

[13] "Amazon web services (aws) - cloud computing services." [Online]. Available: https://pages.awscloud.com/AWS-Innovators-Amazon.html

[14] C. Cox, "Solar," Nov 2021. [Online]. Available: https://www.circuitbasics.com/how-to-use-solar-panels-to-power-the-arduino/

[15] "breadboard and jumper wires." [Online]. Available: https://www.amazon.com/Standard-Jumper-Solderless-Prototype-Breadboard/dp/B07H7V1X7Y

[16] [Online]. Available: https://www.amazon.com/Mmoment-MS11-120GB-Internal-SSD/dp/B07HG81C34

[17] P. P. Ray, "A survey on visual programming languages in internet of things," *Scientific Programming*, vol. 2017, 2017.

[18] I. J. Koenka, J. Sáiz, and P. C. Hauser, "Instrumentino: An open-source modular python framework for controlling arduino based experimental instruments," *Computer Physics Communications*, vol. 185, no. 10, pp. 2724–2729, 2014.

[19] [Online]. Available: http://epa.state.il.us/water/conservation/lake-notes/lake-aeration.pdf