TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES
Ayala Blvd., Ermita, Manila
COLLEGE OF ENGINEERING
ELECTRONICS ENGINEERING DEPARTMENT
electronics@tup.edu.ph

GalluSpy Cam: Real-Time Monitoring System of Body Characteristics and Behavior of Gallus Domesticus using Computer Vision via Internet-of-Things (IoT)

by

Avenido, Jericho O.
Buban, Mark Zian Louize L.
Catudio, Nicole Anne C.
Ruiz, John Vhincent
San Juan, Errol James B.

August 2023

APPROVAL SHEET

This project study entitled "**GALLUSPY CAM: REAL-TIME MONITORING SYSTEM OF BODY CHARACTERISTICS AND BEHAVIOR OF *GALLUS DOMESTICUS* USING COMPUTER VISION VIA INTERNET-OF-THINGS (IOT)",** has been prepared and submitted by the following proponents:

Avenido, Jericho O. | Ruiz, John Vhincent

Buban, Mark Zian Louize L. | San Juan, Errol James B.

Catudio, Nicole Anne C.

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

ENGR. EDGAR A. GALIDO
Project Adviser

ENGR. LEJAN ALFRED C. ENRIQUEZ
Project Co-Adviser

ENGR. CHERRY G. PASCION
Panel Member

ENGR. GILFRED ALLEN M. MADRIGAL
Panel Member

ENGR. ROMEO L. JORDA, Jr.
Panel Member

ENGR. JOHN PETER M. RAMOS
Panel Member

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

ENGR. TIMOTHY M. AMADO
Head, ECE Department

ENGR. NILO M. ARAGO
Dean, College of Engineering

ABSTRACT

The primary aims of the poultry industry have revolved around the augmentation of productivity, the attainment of high-quality products, and the optimization of production costs, achieved through disease control among the fowls. Some criteria that can be used to assess a chick's quality at young age are their actual weight, growth, body temperature, and mortality. This research study aims to develop a monitoring system that utilizes air quality sensors, camera technology, and thermal imaging to monitor the conditions of 45-day-old chicken or Chinese broiler chickens inside the coop. The system utilized computer vision to measure the chickens' everyday weight, behaviors, and body temperatures and underwent trials to determine the efficiency of utilizing the technology to monitor the conditions of chickens ready for consumption. Thus, it would enable farmers to take prompt action to treat or prevent disease outbreaks, reducing the risks of losses and improving the fowls' health. The data generated by the system are transmitted over the internet to a centralized database, where farmers and other stakeholders could also analyze it. The study involved a 5-day observation period to evaluate the accuracy of the prototype. The system's performance parameters included the qualities mentioned. The testing and data gathering showed that the system could record the mortality rate and weight of the fowls, including the ammonia level and room temperature of the coop. In addition, the behaviors and conditions of the fowls could be recognized through implementation of predictive models. Lastly, the actual fowl weight and predicted weight had a mean percentage error of 2.352%, indicating high accuracy of the system's weight prediction.

ACKNOWLEDGEMENT

The researchers wish to extend their sincere gratitude and profound appreciation to all individuals who have made valuable contributions towards the successful culmination of this research endeavor:

Above all, to God, the source of all wisdom, for providing guidance and inspiration throughout this research journey. It is through divine blessings and grace that the researcher was able to navigate challenges, find solutions, and make meaningful contributions to the field of study. The researcher acknowledges and humbly recognizes the divine intervention that played an integral role in the successful completion of this research endeavor;

To Engr. Edgar A. Galido and especially Engr. Lejan Alfred C. Enriquez, the research advisers, for their unwavering support, expert knowledge, invaluable guidance, and profound influence on the development and the quality of this research through their insightful feedback and constructive criticism;

To Engr. Nilo M. Arago, the Capstone 1 & 2 Instructor and Dean of College of Engineering, and Engr. Timothy O. Amado, Instructor in Methods of Research and the Head of Electronics Engineering Department, for their support and collaboration in the study, and for the help they provided throughout brainstorming meetings and illuminating discussions;

To the Panel Members, Engr. John Peter M. Ramos, Engr. Cherry G. Pascion, Engr. Gilfred Allen M. Madrigal, and Engr. Romeo L. Jorda, Jr., for their challenging questions, suggestions and recommendations for the betterment of the study. Also, for their honest criticism and assistance in presenting this study from the topic up to the completion of this research;

To Mr. Wilfredo M. Fernando and family, the owner of the deployment site, for his provision for the access to the poultry farm, his assistance for data gathering and for sharing his knowledge and expertise in the field of poultry;

To the respondents of this study, for their cooperation, time, and willingness in answering the evaluation form provided; and

To their beloved families, for their steadfast financial, emotional, moral, and spiritual support, especially to the family of Mr. John Vhincent Ruiz, who willingly offered their humble home in every overnight and group meeting assembly.

Thank you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

THE PROBLEM AND ITS SETTING

This chapter presents the introduction, background of the study, research gap, research objectives, significance, the scope and limitations of the study.

1.1 Introduction

The broiler sector in the Philippines comprises a proportion of 20% small-scale backyard flocks, consisting of less than 1,000 birds, while the majority of 80% is dominated by commercial farms. As per available data, there were a recorded total of 588 licensed poultry farms and 175 meat processing facilities across the nation. The demand on chicken is expected to increase from 1.4 million metric tons in 2017 to 1.6 million metric tons in 2020 and 1.8 million metric tons in 2023 [1].

The main focal points of the poultry industry in recent times have revolved around disease control, achieving high production levels, ensuring product quality, and maintaining affordable production costs [2]. A healthy 7-day live weight will increase performance indicators like utmost live weight and feed conversion, keeping overall mortality low [3]. At an early age, the quality of a chick can be assessed using various indicators such as live weight, growth rate, crop fill, body temperature, livability, and mortality. These metrics provide valuable insights into the chick's health, development, and overall well-being [4]. A more accurate and current production method can be achieved with the critical knowledge of the chickens' real-time weight, weight gain, average weight, and deviation. Offering an optimized and financially rewarding logistics and slaughter solution, designed to maximize cost-efficiency. Manual measurements and health conditioning are the most prevalent methods used to measure animal growth. This method, however, is time-consuming and challenging, and it may be upsetting for both the chickens and the stockman [5].

Because of the challenges or limitations of current manual measurements, which need livestock to stand on a flat surface with proper posture and use measuring sticks, tape measures, and other equipment, a non-contact method for measuring body dimensions was developed [6]. Anthropometry has recently advanced from flexible tape measurements to automated three-dimensional optical devices that quickly capture hundreds of body surface dimensions [7].

Computer vision has emerged as a viable real-time technology for automating poultry weighing and processing systems. Computer vision systems can be used in the size, mass, volume determination, sorting, and grading of chicken products due to their non-invasive and non-intrusive nature and ability to offer a wide variety of information [8]. Many scientific researchers have already used camera and computer vision technology to monitor broilers since it is a non-intrusive and cost-effective technology that can be automated to work in real-time [9]. One of these is the Microsoft Kinect Camera. Because of its high quality and low cost, the Microsoft Kinect is already generating substantial interest and expanding attention in farm and livestock applications [10].

Structure from Motion (SfM) has gained popularity as one of the most viable 3D reconstruction methods. SfM is a range imaging approach based on photogrammetry that estimates 3D structuring from a collection of 2D photos acquired at various points of view around the body [11].

However, an algorithm that can report real-time health concerns in the chicken coop has been missing in prior works. Infrared thermography (IRT) holds the potential to be seamlessly integrated into automated systems for monitoring animal health and welfare [12]. Prediction of possible diseases and chicken coop temperature can be determined using IRT. Real-time weight statistics as the data collected after acquiring their weight will help to determine whether their feed needs to be increased or decreased. The data can further be utilized to propose dietary changes or to include additional nutrients [13]. Using color cameras or 3D cameras automatically detects animal behavior and analyzes activity [14]. It can assist farmers in more effectively managing their broiler chickens.

1.2 Background of the Study

The significance of behavior as a fundamental criterion for assessing animal welfare is underscored in a study referenced as [15]. It aids in evaluating the health and requirements of the animals. Technological development improves automated and continuous monitoring of poultry behavior, so owners can evaluate and look over to improve welfare. In a study by Li N, the researchers review the newest technologies used for monitoring and detecting the behavior of broilers and laying hens and discuss the potential of establishing a precision livestock farming (PLF) system. The techniques introduced have the potential to serve as an internet-based tool for automated surveillance of broiler behavior, enabling real-time tracking of individual locations and movements, thereby facilitating the assessment of specific behavioral characteristics.

According to a study by Noh et al., in [16], highly pathogenic avian influenza (HPAI) is widely regarded as one of the most devastating ailments affecting poultry. Prompt reporting of any observed indications of outbreaks to the World Organization for Animal Health (WOAH) is of paramount importance. Timely identification of infected birds plays a critical role in curbing the dissemination of the virus, thereby mitigating the economic ramifications. The study's researchers propose the utilization of thermal imaging technology on chickens and ducks as a promising approach for monitoring and swiftly identifying potentially HPAI-infected flocks, facilitating expedited diagnosis of HPAI.

There are currently two primary methodologies employed for assessing the body weight (BW) of livestock as stated in [17]: 1) direct methods encompass the utilization of specialized industrial scales, either partial-weight or full-weight, strategically positioned at designated areas within expansive farms. These scales enable dynamic measurement of livestock weight. These devices exhibit high accuracy and are designed specifically for their intended purpose and the scale of operations. However, their acquisition cost, as well as the recurring expenses related to calibration and maintenance, must be considered alongside challenges associated with their placement in environments characterized by temperature fluctuations. On the other hand, indirect methods involve manual measurements conducted using measuring tapes and tubes, coupled with regression equations capable of establishing correlations between these measurements and body weight (BW). The advent of contactless electro-optical sensors like 2D, 3D, and infrared cameras, coupled with advancements in computer vision (CV) technologies and artificial intelligence (AI) disciplines such as machine learning (ML) and deep learning (DL), has facilitated their widespread application as biometric and morphometric proxies for estimating body weight (BW).

The most precise approach for determining the weight of animals involves employing a weighing scale, as it directly measures the actual weight by placing the animal on the scale. An alternative method relies on estimating weight through empirical relationships established between various traits and body weight. Measurements such as chest girth, withers height, hip height, or the use of electronic scales can be utilized to estimate animal weight. Monitoring the weight of livestock enables assessment of their growth against predetermined benchmarks. In the livestock industry, accurate weight estimation is particularly crucial for cattle and pigs, as they constitute the primary sources of meat products supplied to the market. In consideration, the chickens are also one of the most affordable and famous foods in the Philippines, the weight of the broilers is also vital since that will dictate and measure the amount or price of the chicken in the market [18].

The utilization of a combined approach involving infrared thermal imaging technology and a neural network was employed in [19] to detect and classify the floating range of body surface temperature in laying hens. The testing phase yielded an impressive recognition rate of 97% and facilitated the extraction of temperature data. The study's findings indicate that healthy laying hens exhibit lower temperatures in the head region compared to sick hens, while higher temperatures are observed in the leg area of healthy hens in contrast to sick hens. Additionally, individual sick hens may exhibit even higher temperatures than their healthy counterparts due to pathological reactions.

1.3 Research Gap

Previous research in the field of agriculture and livestock farming has predominantly focused on studying the effects of stress on animals such as pigs, sheep, and cattle. This has primarily been accomplished through manual weighing and assessing overall health conditions. However, there exists a notable dearth of studies and practical applications pertaining to weight prediction in specific chicken species, namely Gallus domesticus (Silkie Chicken) and white broilers. To address this research gap comprehensively, it becomes imperative to consider additional factors that influence the growth and well-being of chickens. In light of this, the researchers involved in this study recognized the importance of developing a weight prediction and health monitoring system specifically tailored for broiler chickens, thereby enhancing the overall research landscape. The implementation of a weight prediction system plays a crucial role in mitigating the risk of chickens having inferior quality and the occurrence of diseases. It also reduces the time consumed and effort by the poultry owners in determining the weight of each chicken and filtering each chicken to monitor their behavior and health condition.

1.4 Research Objectives

To develop a real-time monitoring system that would monitor the air quality in the chicken coop as well as the behavior, temperature, possible diseases and body characteristics of the chickens inside a chicken coop.

In particular, it aims:

1. To develop a monitoring system that would utilize air quality sensor, camera technology and thermal technology for observing the air quality inside the coop, measuring the body characteristics and room temperature of the chicken coop through computer vision, respectively;
2. To employ a predictive model that would help the system determine the behaviors and conditions of *Gallus domesticus*;
3. To develop an IoT-based interface that can be accessed through a webpage where the resulting data of behavior, body characteristics and possible diseases are monitored; and
4. To test and evaluate the output given by the device by comparing it to the results of manual measurements and health conditioning of the chickens.

1.5 Significance of the Study

Poultry meat derived from chickens holds the position of being the most widely consumed in the country. It also provides a good amount of high-quality protein and minerals with low cholesterol levels. In order to provide quality meat for the community, it is vital to consider that the chicken meat gathered from the poultry is safe and free from any form of avian diseases. The utilization of a real-time camera monitoring system can contribute significantly to the efficient maintenance and management of the chicken coop. It could also reduce the costs of maintaining a chicken farm, especially when infected with avian diseases such as bird flu.

Beginning chicken poultry owners will benefit from more efficient monitoring of the chicken farm through various technologies. Implementing a real-time camera monitoring system would potentially reduce chicken fatality. The study might also help in modernizing and innovating chicken farm monitoring. Inclusion of this initiative can be found within Annex 16: Native Chicken and Native Pig, which falls under Section III, Agriculture, Aquatic and Natural Resources (AANR) of the Harmonized National Research and Development Agenda (HNRDA) established by the Department of Science and Technology (DOST).

The study addresses a number of sustainable development objectives, including Goal 2: eliminate hunger, ensure food security, enhance nutrition, and encourage the adoption of sustainable agricultural practices. The implementation of a real-time camera monitoring system in the poultry farm contributes to the production of chicken meat of exceptional quality. As a result, it ensures that chicken meat purchased by citizens of the country is safe from any avian diseases. The second objective is Goal 9: Foster innovation while promoting sustainable industrialization and building resilient infrastructure. The use of a real-time camera monitoring system contributes to an innovative and modern approach to chicken farm monitoring.

Finally, the study will be used as a reference or foundation for future modifications and improvements to the analysis. This project aims to develop a real-time video monitoring system for chicken coops that may be used in future advancements. The following individuals will benefit from the research:

Community - The study will help the community promote sustainable food production.

Poultry Owners - The study will help poultry owners to produce quality chicken meat, prevent chicken fatality, and improve feeding strategies to reduce food and water costs.

Future Researchers - The study will benefit future researchers in laying solid foundations for their initiatives and producing new discoveries and ideas.

1.6 Scope and Limitations

This study will primarily examine the efficiency of using computer vision and artificial intelligence to determine the conditions of 45 days old broiler Chinese chickens or those that are ready for slaughter based on their behavior and physical attributes. The image processing will use physical attributes, including temperature fluctuations, to determine the presence of disease in a chicken, such as fever brought on by the avian influenza virus or weight loss due to fowl cholera or coccidiosis, among others.

The image processing will also determine the behavior such as activity or lameness of the chicken to predict possible health concerns. The associated behavior can also be linked to alterations in maximum surface temperature (MST), which can arise due to heat stress or ventilation issues. There will be an additional capability to forecast potential sales by determining and calculating the weight of the chicken using their body mass index (BMI) through computer vision.

This study does not cover other breeds such as American or European types of chicken. Diseases unrelated to ammonia, temperature, weight, activity, or movement, cannot be detected.

This study will not provide an extensive recommendation or analysis based on the correlation of factors to diagnose a specific disease which the chickens may have acquired.

1.7 Definition of Terms

To enhance comprehension of the study, the subsequent terms were elucidated within the framework of this research.

Anthropometry - refers to the systematic study of human body measurements, with a specific focus on comparative analysis. In relation to chickens, anthropometry involves evaluating the dimensions, proportions, and overall physical and health status of the birds.

Infrared thermography (IRT) - the process involves the detection of infrared energy emanating from a specific object, which is then converted into an apparent temperature. The outcome is presented as an infrared image, captured by a thermal camera.

Structure from Motion (SfM) - is a photogrammetric range imaging technique used for estimating three-dimensional structures from two-dimensional image sequences that could be coupled with local motion signals.

Photogrammetry - imaging science and technology encompass the disciplines of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiation, as well as other phenomena. These practices are employed to extract accurate and valuable information about physical objects and the surrounding environment.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter provides an overview of the relevant literature and studies related to the development of the proposed topic, Real-Time 3D Camera Monitoring of Vital Signs and Body Characteristics of Poultry using Computer Vision.

2.1 Related Literature

2.1.1 Body Characteristics of Chicken

Body measurement of characteristics means statistics like body mass index, body dimension, body conformation, and other the same information. Pattern construction requires accurate body measurements. The accuracy of body mass index, dimension, and confirmation is important. At present, there is currently a transition from traditional manual tape measure measuring to computerized body scanning or photographic methods [20]. Body measurement can be monitored through optical devices which can be accessible at any given place or time.

2.1.1.1 Body Mass Index

The body mass index (BMI) is a quantitative parameter used in medical assessments to determine the proportion of body fat relative to an individual's height and weight [21]. As reported by the Mukono Zonal Agricultural Research and Development Institute [22], male chickens exhibited higher live body weights and other physical measurements compared to females, and these measurements increased with age. On average, mature male chickens weighed approximately 2.11 ± 0.27 kg, while females weighed around 1.48 ± 0.15 kg. With the exception of Body Length and Femur Circumference in females, there were strong positive correlations between body weight and the other parameters. Chest circumference was the most accurate single live weight predictor, followed by Body Length, and Femur Length, with Femur Circumference being the least accurate. In comparison to simple linear regression and polynomials, the power model was the most reliable in the context of live body weight prediction, the utilization of chest circumference as a parameter is explored.

From 7 to 42 days of age, a comprehensive assessment of daily body weight and weekly body length changes in chickens was conducted. The data gathered from this monitoring process was utilized to create a Gompertz growth function, allowing for the estimation of growth parameters. Subsequently, profile analysis was employed to discern variations among the groups. Remarkably, disparities in BMI profiles were observed, suggesting potential distinctions in the feeding practices followed. Consequently, the divergence in BMI across different weeks can be attributed to the specific feeding regimen implemented [23]. Furthermore, conducting weighings of broiler chickens at a higher frequency than the conventional weekly practice provides timely feedback on how feed distribution impacts body weight gains [24].

2.1.1.2 Body Dimension

The body dimensions in chickens encompass various measurements, including height at the withers, chest depth, back height, waist height, and body length [25]. Chickens typically possess a compact, rounded physique, with standard breeds typically standing less than 70 cm (27.6 inches) tall and weighing around 2.6 kg (5.7 pounds). Distinguishing features of roosters and hens include fleshy combs, lobed wattles that hang below the bill, and high-arched tails. Some roosters can even boast tails exceeding 30 cm (12 inches) in length [26]. According to Mushonga [27], weekly increases of 0.47 cm, 0.56 cm, 0.13 cm, 0.26 cm, 1.44 cm, 0.93 cm, 0.95 cm, and 1.15 cm were observed in shank length, keel length, beak length, comb length, chest girth, neck length, wing length, and body length, respectively. Furthermore, a change of 1 cm in body length corresponded to changes in shank length, keel length, beak length, comb length, chest girth, neck length, wing length, and body length. Linear body measurements serve as reliable predictors for determining the body weight of chickens. Shank circumference demonstrated the strongest linear relationship as a predictor, while all other linear body measurements displayed significant effects on the model, including intercepts and slopes of the regression lines [28].

2.1.1.3 Body Conformation

In the evaluation of body conformation in chickens, parameters such as body length, pelvis width, shank length, shank width, keel length, breast width, and breast depth were taken into consideration [29]. Least square means were calculated for chick weight, live body weights (BW), shank length (SL), keel length (KL), and breast angle (BA). According to Rahim's study [30], significant gender-based differences were observed in BA at the 4th week, SL at the 12th and 16th weeks, and KL at the 16th week, with males displaying notably higher estimates compared to females.

2.1.2 Behaviors of Chicken

Cold weather poses relatively fewer challenges for chickens in comparison to heat. Due to their inability to perspire, chickens employ alternative cooling mechanisms such as dipping their beaks in cold water or flapping their wings to dry their feathers. Panting is also observed as a behavior exhibited by chickens when they are in a state of distress and actively trying to cool themselves off [33].

Chickens demonstrate diverse behaviors associated with territory, mating, distress, danger or fear, contentment, and food exploration [31]. The study also uncovered the chickens' preferred behavioral sequences, including comfort-related behaviors. Noteworthy discussions revolve around perching, locomotion activities, and aggressive behaviors. These behaviors are prevalent in commercial broiler and broiler breeder chickens, making their monitoring essential for assessing poultry welfare [32].

2.1.2.1 Wing Spreading

Because chickens don't perspire, they are unable to expel body heat through their skin pores like humans can. Your hens will cool down by dipping their beaks in water and swinging their wings to allow air to circulate through their feathers when it's hot outside. Your hens are healthy if you see them flapping their wings in the summer. They desire to chill off. On hot days, chickens pant and breathe quickly to expel heat from their bodies in addition to flapping their wings [34].

2.1.2.2 Hyperventilating

Excessive ventilation of the lungs, above what is required to achieve normal arterial blood gasses, is referred to as hyperventilation. When hyperventilation happens on a regular basis or in recurrent bouts and is accompanied by physical or psychological symptoms [35]. Chickens pant by opening their beaks and inhaling quickly. Chickens release heat through panting to dissipate internal heat, similar to dogs. As early symptoms of heat stress, look for panting (beaks open) and rapid breathing [36].

2.1.2.3 Inactivity

The affected chicks are inactive. Some chicks will either refuse to walk or walk on their hocks. Many chicks will die by lying with both feet out to one side (prostrate) in advanced cases. In an affected flock, all stages (dullness, tremors, prostration) are frequently visible [37]. Selection for rapid development has resulted in broiler chickens that are heavier and more efficient, but it has also resulted in health and welfare issues that may be caused or worsened by inactivity. Rapid growth may limit the performance of motivated actions, but enrichment may boost these behaviors and overall activity [38].

2.1.2.4 Scratching

Be alert for lice in your flock if you notice your chickens scratching on themselves rather than the ground. External parasites called lice eat the skin and feathers of hens. As they migrate between the feathers, they are tiny but noticeable. At the base of the feather shafts lie their pale egg masses [39].

2.1.3 Air Quality

Intensive poultry farming degrades indoor air quality and releases air pollutants and other elements that affect the global atmosphere. The poultry feces, which is high in organic matter and nitrogen, becomes a source of gaseous emissions inside the poultry house. By regularly monitoring the air quality within the poultry house, waste clearance may be scheduled to ensure that both the workers and the birds are breathing clean air.[40] Air emissions from poultry and livestock production can include dust or particulate matter, odors, endotoxins, methane, hydrogen sulfide, carbon dioxide, mineral emissions such as P, N, Zn, and Cu, and nitrogenous compounds such as ammonia. Ammonia emissions have the potential to cause wet and dry deposition as well as surface and groundwater contamination.[41]

2.1.4 Body Characteristics Measuring Cameras

Body characteristics measuring cameras are cameras capable of non contact measuring the body dimensions. RGB-D and Kinect are cameras that are used to measure different body characteristics.

2.1.4.1 RGB-D Camera

RGB-D Sensors are a form of depth sensor that acts in accordance with an RGB (red, green, and blue color) sensor camera. They can add depth information (related to the proximity to the sensor) to a standard image on a per-pixel basis. In the past few years, depth sensors have motivated researchers in the area of computer vision and computer graphics to develop new RGB-D image-based solutions. Many difficult tasks, such as object recognition, scene interpretation, pose estimation, visual tracking, semantic segmentation, shape analysis, image-based rendering, and 3D reconstruction, may benefit from the use of depth information [42].

2.1.4.2 Microsoft Lifecam

High Definition video (1.3 megapixels)3, 3x digital zoom, and the best still photography on the market are all features of the brand-new Microsoft LifeCam VX-6000, a top-tier wired webcam. Up to three individuals can join the chat thanks to the 71-degree wide-angle lens. High Definition still photography (1.3 megapixels interpolated) and stunning video (640x480 pixels) are also features of the Microsoft LifeCam VX-3000. [43].

2.1.5 Behavior Monitoring Camera

Fancom BV, a company that specializes in the automation of livestock facilities, developed and manufactures the camera system known as EYeNamic. It has three cameras that can continuously record the behavior of chickens on the top of the broiler house. Through the use of eYeNamic, it is possible to process the photos and measure the distribution and activity of the animals, which can be thought of as important indicators of animal welfare [44].

2.1.6 Image processing

Image processing is a technique that helps to improve image quality with the utilization of different methods of processing images. There are also several applications of image processing that are being studied nowadays. There have been researches conducted that involve image processing with calculating processes.

2.1.6.1 Pixel Value Calculation

Generally, to measure a person's BMI, one must measure necessary parameters such as the height and the weight of the person. The next step is to proceed with the calculations to determine the condition of the subject's body mass index (BMI). The study [45] proposed the use of image processing to determine the BMI of a person. They have proposed a method where in the first step the camera captures the image of the human in a standing position from a distance of 300 cm from the human and is elevated by 80 cm from the floor. After the capture of the subject, their program would perform an approximation on a human's height and weight using pixel value calculation. The program filters out the image where it removes the background of the image for better approximation. Based on the results, the average error rate of the subject's height and weight are 4.1% and 8.6% respectively.

2.1.6.2 Artificial Neural Network (ANN)

Image processing has also been used in the agriculture sector, specifically in detecting disease and grading fruit. The increased demand to effectively grow a crop and increase its yield made the monitoring of the crops much more important. In this study [46] image processing has been used to monitor the condition of the crops, from when it is planted up to when it is harvested. The researchers made use of the concept of artificial neural network. The researchers decided to select three diseases of grapes and two diseases of apples. They have made a system where it uses two image databases, to achieve its purpose, one for training already of stored disease images and one for implementing query images. For the weight adjustment of the training database, the concept of back propagation was used. The images were distributed to their particular categories based on three feature vectors, specifically, color, texture, and morphology. Based on these feature vectors, morphology was the one that gave 90% correct results.

The paper demonstrated the effectiveness of algorithms for the spread of disease and counting of mango. The practical implementation of neural networks was made possible with the use of MATLAB.

2.1.6.3 Statistical Image Analysis

The study [47] states that abnormal behavior of animals and declined rate of growth could be an indication of an unwanted situation, namely diseases, technical malfunction in feeding and drinking lines and subpar management procedures. It has always been important to detect problems early in order to avoid harming the welfare or the production of broilers. The paper introduced an automated technique to detect such problems with the use of cameras and an image analysis software. Three top view cameras were set up on the ridge of a commercial broiler house that has 28000 animals that are monitored continuously and the images captured were to be translated into animal distribution index with an interval of 5 minutes by the analysis software. Based on the translated data, a real-time model was created for the sake of predicting the normal distribution index as a feedback to the light input. With the use of the model, it was made possible to create an online prediction in real-time as to produce an alarm whenever unusual animal behavior occurred apart from the prediction. The results from the said method was able to report 95.24% of events in real-time, which tells that there is a high potential when using automatic monitor tools for broiler production over a complete growing period.

2.1.6.4 Linear Regression Method with the use of NIR camera

Traditionally, classifying table fruits is done manually by hand, and this causes faulty classifications.[49] This process could be done automatically with the use of machines and is important in terms of speeding up the process, reducing the costs and minimizing the errors. The study used image processing techniques in order to estimate weight and diameter on "Starking" type apples. Initially, 50 pictures were captured using NIR camera accompanied by an 830 nm long pass filter. It was then followed by performing edge detection algorithms and morphological operations on the images in order to obtain the image boundaries. Based from the information obtained from the binary images, the diameter and area information were used as attributes which are then given as input to the Linear Regression method. The results showed that 93% of the diameters of the apples and 96.5% of the weights could be estimated.

2.1.7 Microprocessor

A microprocessor, also known as a CPU or central processing unit, is a single-chip computing engine that performs all computations [50]. A camera with a microprocessor capable of receiving varied photographic data via interrupt processing [51].

2.1.7.1 Raspberry Pi

The Raspberry Pi is a computer that runs on a microprocessor. Several peripherals can be connected, including a monitor (through HDMI or AV Port), mouse and keyboard (via USB), internet connection (by Ethernet or Wi-Fi), and the addition of a camera (through the dedicated Camera Interface) [52].

2.1.8 Data Communications

The transmission of this digital data between two or more computers is known as data communications. A computer network, also known as a data network, is a type of telecommunication network that allows computers to communicate with each other. Cable or wireless media is used to establish a physical link between networked computing devices [53].

2.1.8.1 Web Application

A Web application (Web app) is a software that is stored on a remote server and distributed via the Internet using a browser interface. Web applications can be designed for a wide variety of fields and utilized by anyone, from companies to individuals, for a variety of reasons. Some Web apps are only accessible through a certain browser; however, the majority are accessible through any browser [54].

2.1.8.1.1 Internet of Things

The Internet of Things (IoT) is a network of real-world items, or "things," that have been outfitted with sensors, software, and other technologies to communicate and exchange data with other systems and equipment on the internet [55]. According to Dong Ma [56], the Internet of Things (IoT) is a program that aims to connect and integrate the physical and virtual worlds. It is the third wave of the IT industry revolution, and it represents the future networking trend.

2.1.8.2 Web Hosting

Web hosting is a service that provides internet storage space for websites. The World Wide Web provides access to these web pages. Web hosts are companies that provide website hosting services. The servers that host the website are turned on 24 hours a day, seven days a week. Web hosting companies are in charge of these servers. Every server has a unique IP address. Webmasters point their domain name to the IP address of the server where their website is stored because IP addresses are difficult to remember [57].

2.1.8.2.1 Shared Hosting

Shared hosting, also known as virtual hosting, is a method of hosting your website on a single physical server with other websites. There are software applications on the server that make administering and accessing your website simple [58]. Shared hosting is suitable for small to medium-sized businesses or organizations, personal projects (blogs or portfolios) with less traffic requirements, and people who want to create websites but don't have the time or technical skills to manage their servers [59].

2.1.8.2.2 Virtual Private Server Hosting

A virtual private server, or VPS, is a type of multi-tenant cloud hosting in which virtualized server resources are made available to an end user via the internet through a cloud or hosting provider [60]. VPS provides complete system control, root access, scalability, dedicated technical support, and high performance. Every business owner may benefit from a VPS [61].

2.1.8.2.3 Cloud Hosting

Cloud hosting is a type of server and network infrastructure in which a single physical server is divided into multiple virtual servers using software. Sometimes, these devices are referred to as virtual machines or VMs [62]. Cloud hosting enables programs and websites to access cloud resources. Solutions are not implemented on a single server, unlike traditional hosting. Instead, the application or website is hosted by a network of connected virtual and physical cloud servers, which provides greater flexibility and scalability [63].

2.1.8.3 Web Programming Language

Web programming covers all areas of Web development, including Web content, Web client and server scripting, and network security [66]. Programming languages are chosen by developers based on project objectives, learning curve, and reliability, among other factors. Python, C, and Java are three of the world's top five programming languages [67].

2.1.8.3.1 HTML

HyperText Markup Language, or HTML. It is a common markup language used to create web pages. Using HTML elements, or the components that make up a web page, like tags and attributes, it enables the construction and organization of sections, paragraphs, and connections. [68]. This program employs web-based HTML and VNC

(Virtual Network Computing) methods to connect to the lab computer. The ThinVNC server is installed on the server machine, and the user, password, and port are configured. Because it only requires a web browser that supports HTML 5 (such Chrome, Firefox, or Edge) and does not require the installation of any browser plugins or RDP, ThinVNC serves as a solution that makes it simple for users to access the lab from a tablet or smartphone. [69].

2.2 Related Studies

2.2.1 Local Studies

2.2.1.1 Weight Estimation in Fisheries

2.2.1.1.1 Weight Prediction System for Nile Tilapia using Image Processing and Predictive Analysis

The study [70] used a Raspberry Pi microcontroller for Hough gradient method-based weight prediction system for Nile Tilapia (*Oreochromis niloticus*), equipped with low-cost USB cameras to monitor its growth by its length and weight. The fish become prone to stress when captured just to its length and weight and monitor the growth, thus, a weight prediction application is proposed. It is also aided by t-test to further give more accurate weight estimation. Fish growth rose by 47.88% using the weight prediction application. The manual weighing of aquatic animals could potentially harm them. As observed, the development of the application has seen to be more advantageous than manual weighing since it monitors the fish growth without threatening them or worse, leading to their fatality.

2.2.2 Foreign Studies

2.2.2.1 Weight Estimation in Livestock using Different Techniques

2.2.2.1.1 Weight Estimation based on Neural Network with Regression in Barqi Breed Sheep

Bhatt in [71] used a single image held from a normal camera with 3 channels (RGB) to gather the linear dimensions from the image. The researcher also incorporated information about sheep's gender and age to improve the results. The architectures are based on Convolution Neural Nets, a deep learning technique to solve computer vision. This SegNet model uses an encoder-decoder architecture to execute segmentation; it also uses three dimensions to coincide with the height, width and the number of classes in the image. Segmentation is an unavoidable part of any algorithm in computer vision, there are many algorithms for segmentation. In this study the researcher used unsupervised image segmentation using autoencoder.

2.2.2.1.2 Weight Estimation using a Single 3D Video Capture Device in Cattle

A study conducted by Hansen [72], implemented a low-cost automated system with the help of 3D imaging technology that was used in cattle monitoring and assessing the animals condition data such as weight, body condition and lameness. With this system it benefits the cattleman as the milk production increases and reduces the feed and veterinary cost, while for the animal it reduces the discomfort and pain, it also increases their lifespan. The system monitors the whole herd daily, gathering the needed and real-time information to boost every cattle condition and to classify the actions or tasks to prevent future problems. The study achieved a classification accuracy of 83%, the automatic extraction of the spine based on accurate depth imagery was not used or practiced before.

2.2.2.1.3 Deep Learning Techniques for Beef Cattle Body Weight Prediction

Gjergji et. al [73] used varieties of deep learning models performance to predict cattle weight and compared which model had the most accurate result. Convolutional neural networks, RNN/CNN networks, Recurrent Attention Models, and Recurrent Attention Models with Convolutional Neural Networks were utilized in the study, and convolutional neural networks outperformed them all. The MAE of the best model is 23.19 kg on average. This is almost half the inaccuracy of previous top linear regression models, which had a 38.46 kg error. The weight estimation for the beef cattle was difficult since there is loss of information in object shape when going from 3-D space to 2-D photos. In order to minimize the weight estimation error for cattle weight, the data points must be considered and a better deep learning model is recommended.

2.2.2.1.4 Algorithm of Sheep Body Dimension Measurement and its Applications Based on Image Analysis

The study [74] used image analysis to measure the sheep body dimensions, such as body height, rump height, body length, chest depth, chest width, and rump width. The body dimension measurement is necessary in sheep breeding to reflect its growth development, production performance and genetic characteristics. Body dimension measurement using image processing yields more desirable results compared to the manual measurement. However, the researchers must measure both the left and right portions of the sheep to have more precise results. The mobility of the sheep could make the dimension estimation prone to error. Hence, repeated comparison between the image processing and manual measurement could be considered.

2.2.2.1.5 An Intelligent Pig Weights Estimate Method Based on Deep Learning in Sow Stall Environments

Cang et. al [75] applied artificial intelligence techniques through the use of deep neural networks to estimate pig weights in saw stalls. Cameras are placed to monitor the back of pigs in top-view depth images. The weight estimation program is Faster-RCNN network-based with an added regressive branch that performs both the pig recognition and pig weight estimation at the same time. The actual weight of the 20 pigs are first measured, ranging from 159.27 kg to 167.27 kg. The weight estimation yielded to an average error of less than 1%, thus, it gave more accurate weight prediction for the pigs. The movement of the pigs in the control area could affect its accuracy.

2.2.2.2 Weight Estimation in Poultry

2.2.2.2.1 Study on Duck Weight Estimation by using Image Processing

Lieng and Sangpradit [76] used LabVIEW with the vision builder to explore the possibility of estimating the weight of the ducks with the help of image processing. The experiment is done in a smart commercial farm in Ongkharak district, Nakhonnayuk Province, Thailand. Each duck image is captured using a SONY digital camera in JPEG format. Each duck had an actual average weight of 3.255 kg while the estimated average weight using image processing was approximately 3.152 kg. The image processing resulted in an average weight error of 103.1 grams, equivalent to 3.27% error, compared to the actual digital duck weight. Thus, the system was found to be accurate enough for duck weight estimation. The enclosed weighing room secured the ducks to avoid unnecessary mobility that could potentially make the weight prediction prone to errors.

2.3 Summary of Related Studies

Table 2.1 Summary of Related Studies

Research Title	Author/Year	Methodology	Findings / Recommendation
Weight Prediction System for Nile Tilapia using Image Processing and Predictive Analysis	Tolentino, Lean Karlo S.; De Pedro, Celline P.; Icamina, Jatt D.; Navarro, John Benjamin E.; Salvacion, Luigi James D.; Sobrevilla, Gian Carlo D.; Villanueva, Apolo A.; Amado, Timothy M.; Padilla, Maria VictoriaAllen M.(2020) C.; Madrigal, Gilfred	The proponents described a low-cost monitoring and Hough gradient method-based weight prediction system for Nile Tilapia (*Oreochromis niloticus*) using a Raspberry Pi microcontroller and two low-cost USB cameras.	The weight predicting system for Nile Tilapia (*Oreochromis niloticus*) was successful without human intervention, as it could lead the fish to undergo stress. The growth of the fish increased by a significant amount. Monitoring the movement of Nile Tilapia could be recommended to further detect unusual movements.
Deep Learning Techniques for Beef Cattle Body Weight Prediction	Gjergji, M; Weber, M; Silva, L.; Gomes, R.; de Araujo, T.; Pistori, H; Alvarez, M. (2020)	The proponents had proposed a low-cost automated system for the unobtrusive and continuous welfare monitoring of dairy cattle on the farm. It made use of 3D imaging technology to obtain varying forms of related animal condition data (body condition, lameness, and weight), concurrently with the use of a single device. Convolutional neural networks, RNN/CNN networks, Recurrent Attention Models, and Recurrent Attention Models with Convolutional Neural Networks were used to predict the cattle weight and compare the effectiveness of all methods.	Using all the methods, only the convolutional neural networks method gave the most accurate weight prediction.The weight prediction for the cattle was found out to be difficult due to loss of information in object shape caused by the transition of 3-D space to 2-D photos.

Algorithm of Sheep Body Dimension Measurement and its Applications Based on Image Analysis	Lina Zhanga, Pei Wu, Tana Wuyuna, Xinhua Jiangc, Chuanzhong Xuana, Yanhua Maa (2018)	The method used relied on computer-assisted visual image capture in a position-limit apparatus, as well as an automatic foreground area extraction algorithm is known as simple linear iterative clustering (SLIC) SuperPixels and Fuzzy c-means (FCM) clustering, a center line of flexible symmetrical body extraction algorithms, and measuring point extraction algorithms.	The dimension estimation for the sheep was found to be accurate compared to the actual measured dimensions. Thus, the workload for manual measurement could be reduced, as well as the sheep's stressors. Both the right and left portions of the sheep must be measured to improve the system's accuracy. In addition, the repeated comparison of the actual measured and estimated weights of the sheep is necessary.
An Intelligent Pig Weights Estimate Method Based on Deep Learning in Sow Stall Environments	Cang, Yang; He, Hengxiang; Qiao, Yulong (2019)	The weight estimation program is based on the Faster-RCNN network-based program with an added regressive branch. It simultaneously performs pig location, pig recognition, and pig weight estimation.	The weight of each pig was accurately predicted, having an average error of less than 1%. However, the sudden movements of the pigs could affect the effectiveness of weight estimation.

Study on Duck Weight Estimation by using Image Processing	Lieng, Phatpisey; Sangpradit, Kiattisak (2020)	The proponents used LabVIEW with the vision builder for estimating the weight of the ducks. Duck images are captured using a SONY digital camera in JPEG format.	The system had accurately predicted the weight of 10 ducks as compared to each duck's manually measured weight. The image processing resulted in a 3.27% error compared to the actual digital duck weight. The proponents addressed the potential significant errors in weight prediction by weighing the ducks in an enclosed weighing chamber to prevent movements that may lead to mistakes in weight estimation.

2.4 Comparison Table

Table 2.2 Comparison of Other Study to GalluSpy Cam

	IJACSA [70]	IEEE[73]	ELSEVIER [74]	IEEE[75]	EDPS [76]	Our / This work
Year	2020	2020	2018	2019	2020	2023
Method	Image Processing and Predictive Analysis	Computer Vision and Image Processing	Computer Vision and Image Processing	Computer Vision and Image Processing	Computer Vision and Image Processing	Computer Vision and Image Processing
% Error	2.82%	N/A	2.30%	1.60%	3.27%	2.35%
Quantity	10 tilapia	7 cows	27 sheep	N/A	10 ducks	10 chickens
Neural Network	Artificial Neural Network (ANN)	Recurrent and Convolutional Neural Networks (RNN/CNN)	Convolutional Neural Network (CNN)	Region-based Convolutional Neural Network (R-CNN)	Convolutional Neural Network (CNN)	Artificial Neural Network (ANN)

CHAPTER 3

METHODOLOGY

This chapter discusses the research methodology and data analysis techniques employed to address the research objectives.

3.1 Research Design

3.1.1 Developmental Research

The researchers will design, assess, and observe the system's results and functionality using developmental research design. A quantitative research method will be applied to this study as the researchers search to answer quantitative questions and parameters that can be measured. The researchers will use the testing and evaluation processes to collect, record, and evaluate data in order to reach the intended outcomes, which will be directed by the user and system requirements. The participants in this study will be poultry owners to be selected from various poultry farms.

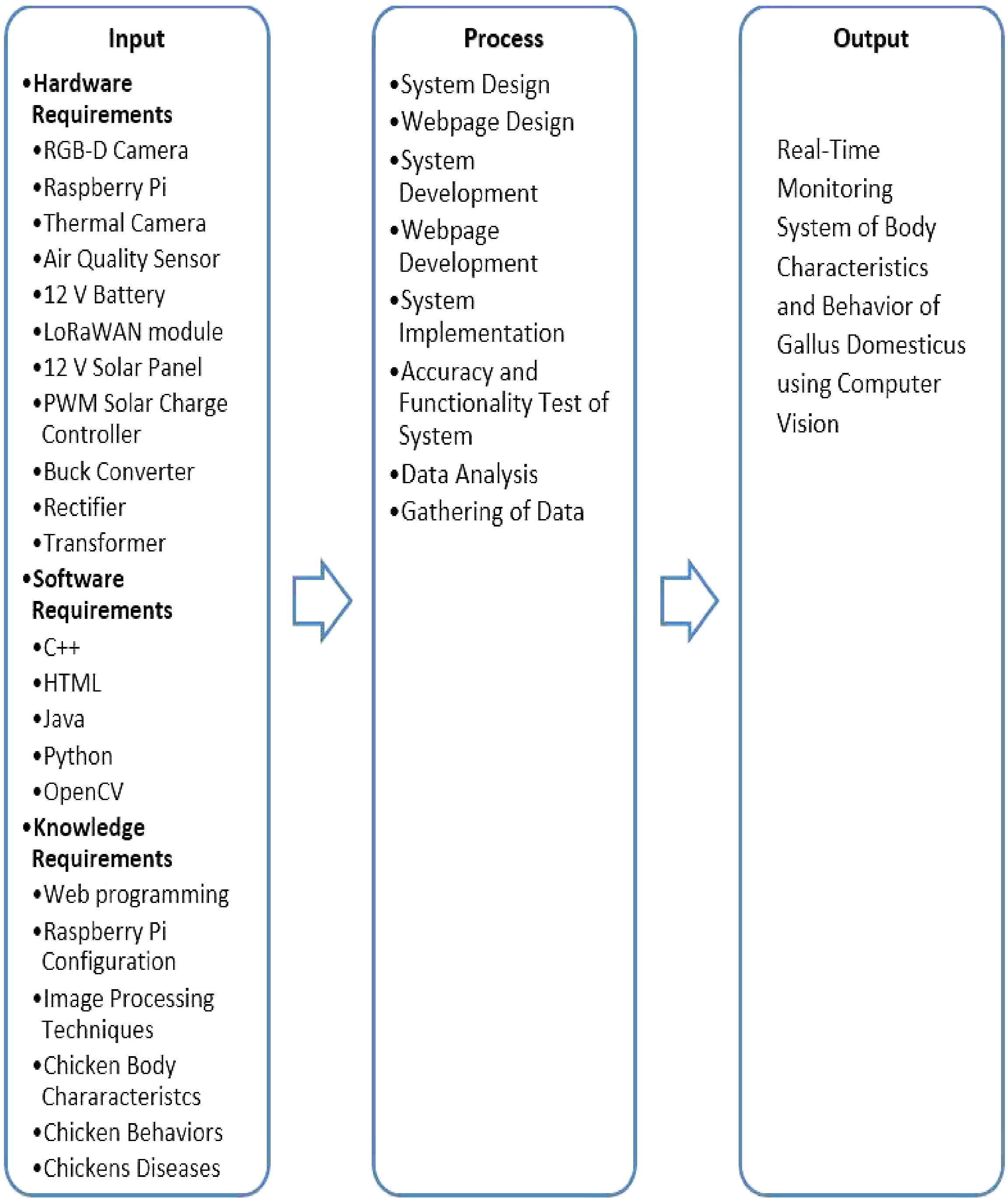


Figure 3.1. Input Process Output Model (IPO Model)

Figure 3.1 shows the study's IPO model, in which the input block contains all the data required to proceed as well as the requirements for both the hardware and software areas. The main focus of the software requirements is expertise in computer-vision programming, which includes data and image processing. On the other hand, hardware requirements are composed of all the additional components required to create the real-time monitoring camera. Having said that, the project's input will come via the Raspberry Pi connected to a RGB-D camera, thermal camera, and air quality sensor. The Raspberry Pi is equipped with calibration sets for the output of an image processing algorithm and a predictive model to predict possible diseases and predict chicken weight. These models are based on data previously obtained through prevalent methods to serve as test data for weighing and health condition chickens inside a poultry house.

Using LoRaWAN remote gateway, the processed data will be transmitted to the cloud and made accessible on a website for monitoring along with real-time growth rate, behavioral changes, and possible diseases that will show whether the chicken acquired a disease or the poultry house contains methane.

3.2 Research Process Flow

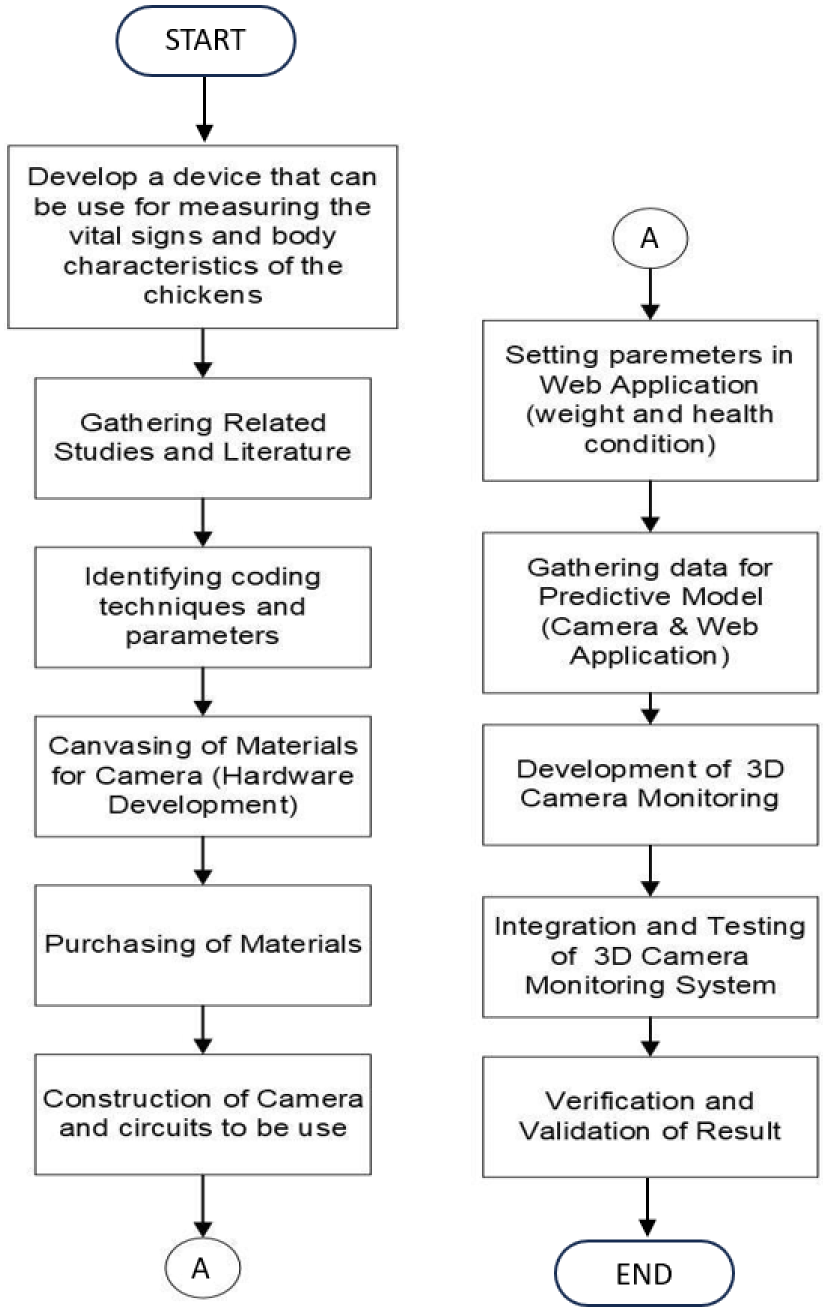


Figure 3.2. Process Flow

Figure 3.2 shows the process flow for conducting research occurred chronologically during the study. The method began with conceptualizing how real-time monitoring of *Gallus domesticus* came to be the primary focus of the study, then with conducting a study on those who are particularly concerned in this area. The researchers then looked for projects that had already been completed in order to determine what needed to be changed or improved. Prior to building the setup, the weight gathering and computer-vision programming were completed.

After completing the required programming, the team started creating the real *Gallus domesticus* surveillance camera. The subsequent steps involve assembling the devices and connecting every component required for deployment. After a thorough evaluation of the entire 3D camera monitoring system and data collection, the results were verified.

3.3 Block Diagram

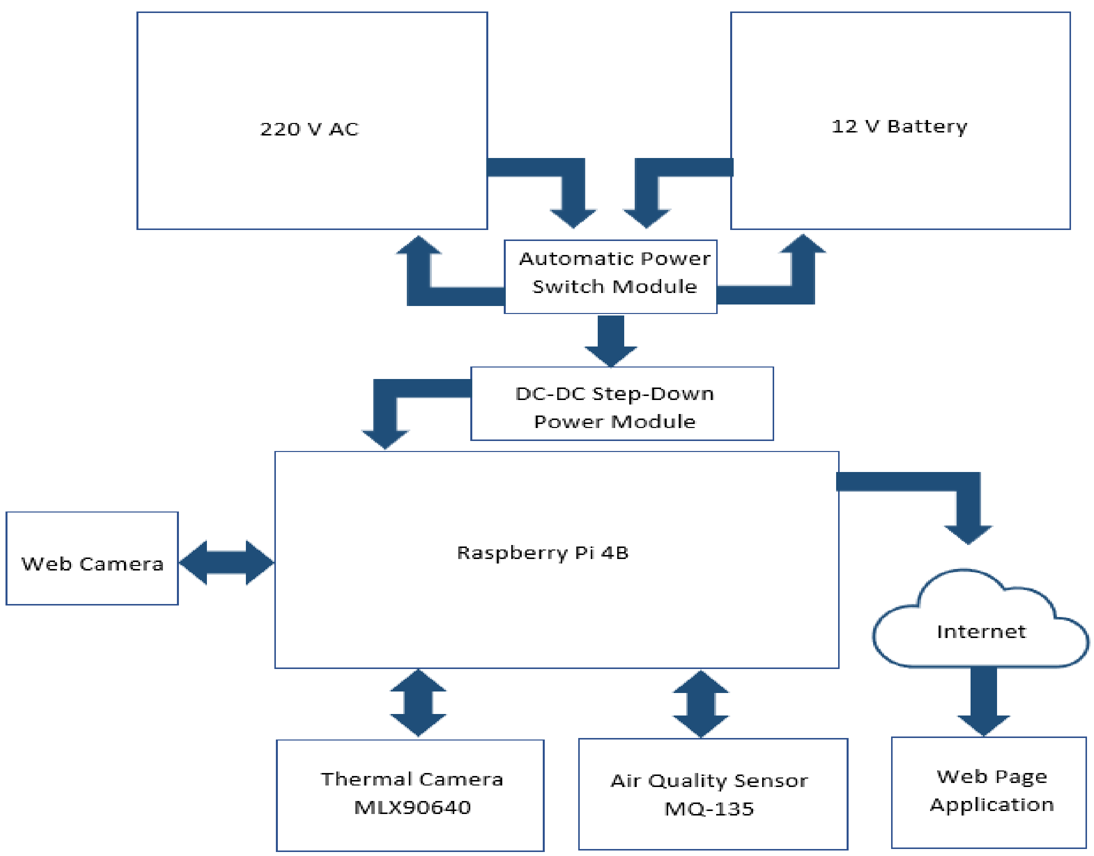


Figure 3.3. Block Diagram of the Real-Time Camera

Figure 3.3 shows the block diagram containing all the essential components and processes needed to make the real-time camera. It indicates the process and how they are all connected or attached to one another. 220 V AC and 12 V Battery serves as the power source, both of the source is connected to the Automatic Power Switch Module that will determine what to use when one power source is not available. The automatic power switch module is connected to the DC-DC Step-Down Module which reduces the input voltage. DC-DC Step-Down Module was connected to Raspberry Pi for power supply. The Webcam or the Microsoft Lifecam, Thermal Camera (MLX90640), and Air quality sensor (MQ-135) gather information and send the data to the Raspberry Pi. Afterwards, the data gathered would then be displayed in the web application.

3.4 Hardware Development

This section presents the different methodologies in the development of the hardware prototype and the concepts and parameters for the real-time monitoring system of body characteristics and behavior. The hardware requirements of the prototype are discussed in this section.

3.4.1 Materials and Equipments

3.4.1.1 Microsoft Lifecam

Microsoft Lifecam is a high definition webcam with a 1280 x 720 resolution allowing for 720p HD video chat. The integrated CMOS sensor helps reduce the blooming effect to avoid overloading of the light sensor which can cause bleeding of light between pixels. As a result, users can take advantage of higher picture quality with reduced bleeding of the light source. The 68.5° field of view makes it easy to capture wide angle pictures and video, while the 360° rotation capability makes swiveling the camera simple. With a wideband microphone you have the ability to capture quality audio when chatting and recording video. It also uses CMOS sensor technology with Motion Video: 1280 X 720 pixel resolution and Still Image: 1280 X 800 pixel resolution = 1,024,000 pixels with up to 30 frames per second imaging rate. The imaging features consist of digital pan, digital tilt, vertical tilt, swivel pan, and 4x digital zoom; fixed focus from 0.3m to 1.5m; True Color - Automatic image adjustment with manual override; 16:9 widescreen; 24-bit color depth.

Figure 3.4 Microsoft Lifecam

3.4.1.2 Microprocessor

Raspberry Pi is a microcomputer based board that acts as a computer, it is also called General Purpose input/output pins that can be used to interact with keyboards, mouse, monitors, camera, etc. Raspberry Pi is used in several applications, from building amazing gadgets and games to monitoring and surveillance devices. The ideal RAM of raspberry pi for this project is 4GB or higher.

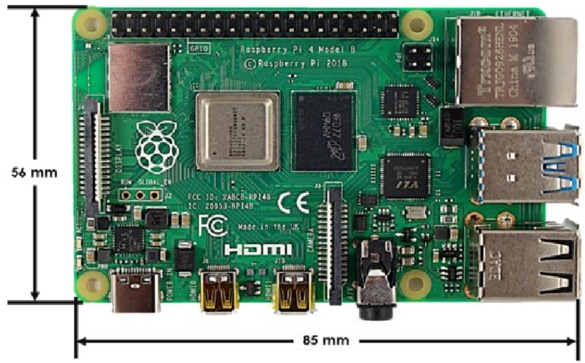


Figure 3.5 Raspberry PI

3.4.1.3 Thermal Camera

A thermal camera uses the infrared radiation that the subject emits to capture and create an image of the subject. The temperature of the object is displayed in the produced image. The Adafruit MLX90640 Thermal Camera is a compact infrared imaging sensor that can detect and visualize temperature variations in its field of view of 55 degrees horizontally and 35 degrees vertically. The camera has a resolution of 32x24 pixels, which means it can measure temperatures at 768 different points within its field of view.

Figure 3.6 Thermal Camera

3.4.1.4 Air Quality Sensor

Air quality sensors are devices used to detect contaminants in the air. This includes substances that could be hazardous to human health, such as particles, contaminants, and noxious gasses. An MQ135 air quality sensor is one type of MQ gas sensor used to detect, measure, and monitor a wide range of gases present in air like ammonia, alcohol, benzene, smoke, carbon dioxide, etc. It operates at a 5V supply with 150mA consumption. Preheating of 20 seconds is required before the operation, to obtain the accurate output. It is highly sensitive to NH3, NOx, CO2, benzene, smoke, and other dangerous gases in the atmosphere. This sensor has a detection range of 10～1000ppm (ammonia gas, toluene, Hydrogen, smoke) and it can detect gases within a range of a few meters.

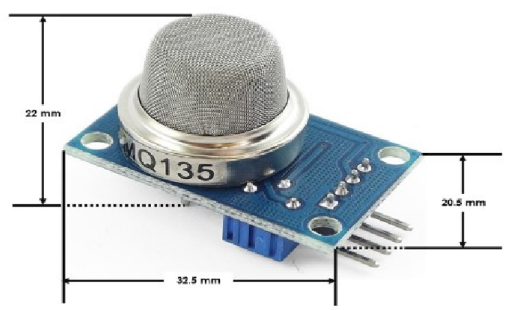


Figure 3.7 Air Quality Sensor

3.4.1.5 12 V Battery

A 12-volt battery is a battery utilized in particular electronic applications. The 12 volt battery is one of the battery kinds that takes on a flexibility and performance appearance depending on what it is used for. It is among the most diverse batteries in certain ways.

Figure 3.8 12-V Battery

3.4.1.6 XY-3606 DC-DC Step-Down Power Module

XY3606 DC-DC Step-Down Power Module is a versatile power supply module with multiple functionalities. It supports a wide input voltage range and provides stable and adjustable output voltage, making it suitable for various electronic applications requiring reliable power regulation.

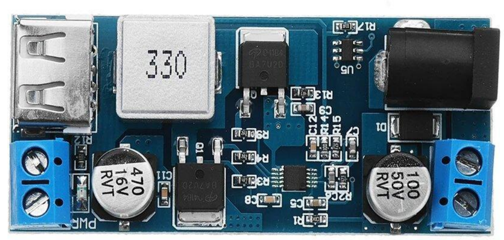
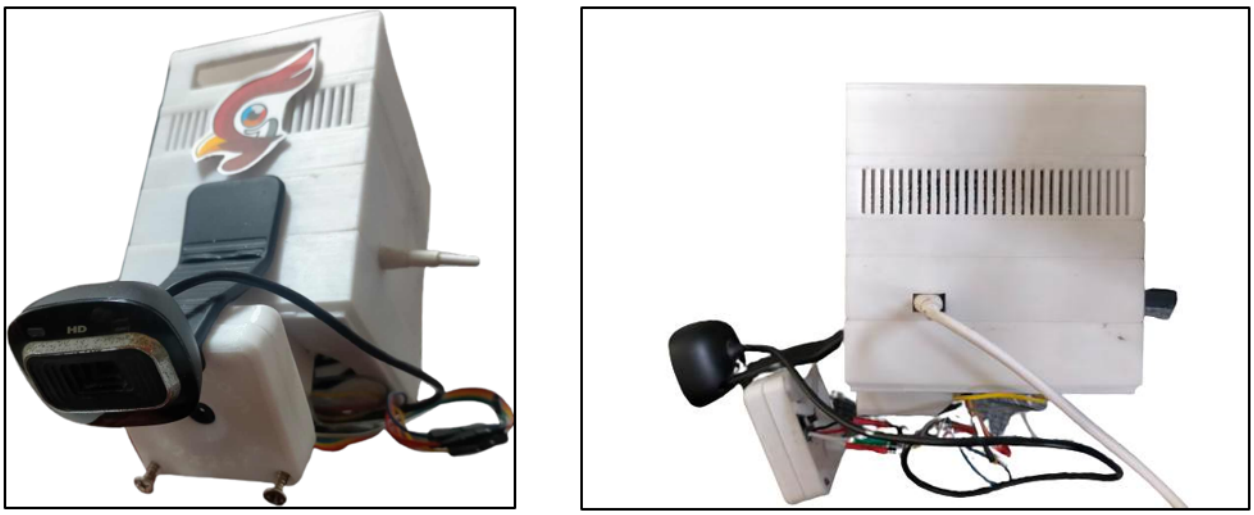


Figure 3.9 XY3606

3.4.1.7 Automatic Power Switch Module - MQ135 Air Quality Sensor

The YX851/DC12V Power Switch Module is an automatic power switching device designed to ensure uninterrupted power supply in emergency or backup power systems. It monitors multiple power sources, such as mains power and backup batteries, and switches between them seamlessly to provide continuous power and efficient charging.

Figure 3.10 YX851/DC12V Power Switch Module



(a) (b)

Figure 3.11 Hardware Design of Monitoring Camera: (a) Front view (b)Left-side view

The researchers will use thermal and RGB-D cameras with air quality sensors. To make the experimental setup fully functioning in the way the researcher intends it to be, several modifications were done. The thermal camera, RGB-D camera and air quality sensor are attached to the system to identify the behavioral changes, body growth changes and possible diseases. Other hardware included Raspberry Pi as a microcontroller for the whole system and LoRaWAN as an IOT gateway. The main power supply for the whole system will be the electrical outlet and solar power technology will be used as a back-up power supply in case of power outage.

3.5 Software Development

The software development for the Real-Time Monitoring System of Body Characteristics and Behavior of *Gallus domesticus* using computer vision will use the incremental model of software development.

3.5.1 System Development of Real-Time Monitoring System of Body Characteristics and Behavior of *Gallus domesticus*

This section presents the flowchart for the system of the prototype which discusses the flow of operation of the system as a whole.

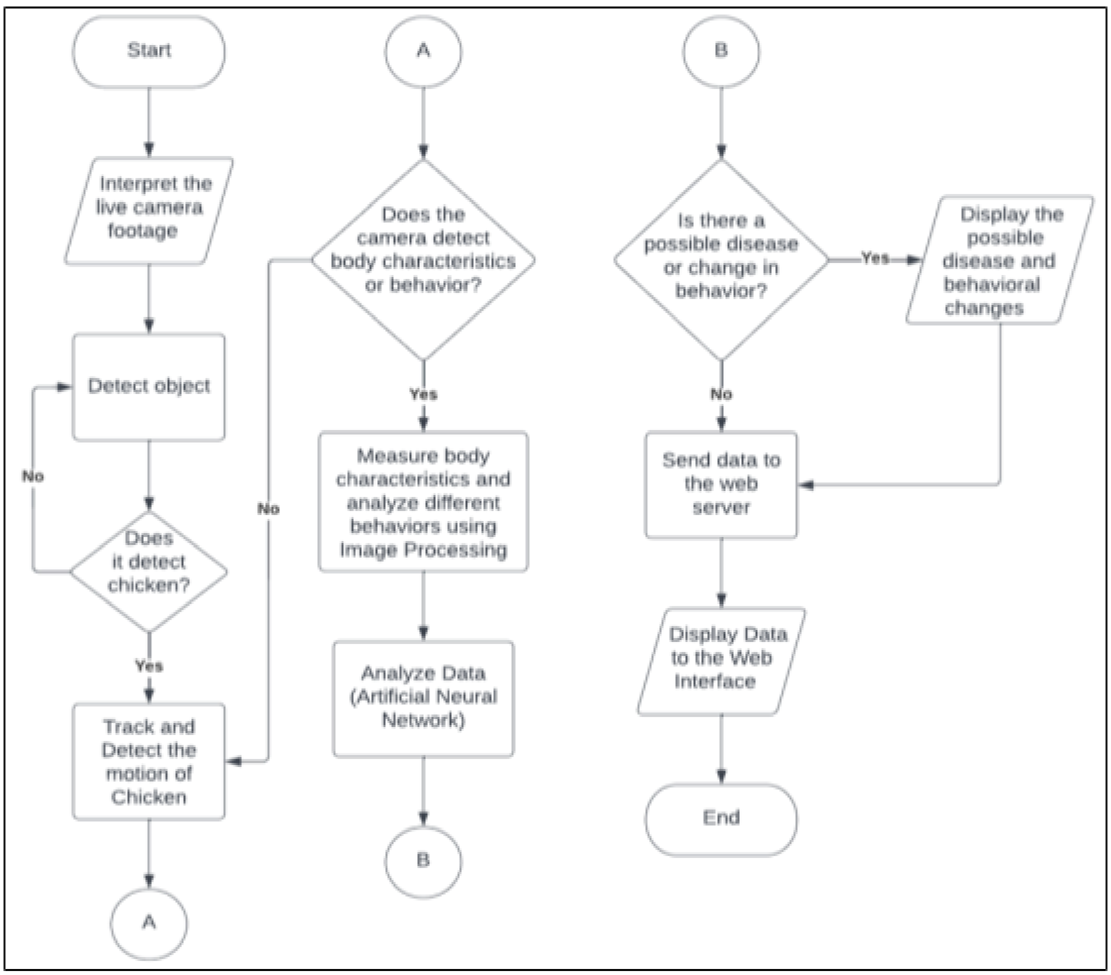


Figure 3.12 System Flowchart Diagram

Figure 3.12 shows the flowchart of the system's software. The system will load first the stereo calibration of the camera and will proceed to interpret the live camera footage inside the coop. It will create stereo images and at the same time it will detect the chickens. It will process the image captured while calculating the pixel value of the image. It will proceed to return the pixel value and will start motion tracking of the chickens. After the motion tracking, it will determine if the camera detects the body characteristics or behavior. If not, it will detect the chickens again and repeat the process until motion tracking. If yes, it will proceed to measure the body characteristics and analyze the different behaviors. The gathered data will be sent to the Raspberry Pi and will be analyzed. It will determine if there are possible diseases in the analyzed data or change in behavior. If yes, it will proceed to display the results and send it to the web server. If not, it won't display any changes and still proceed to the web server. The data received will be displayed in the web server interface.

3.5.1.1 Predictive Model for Behavior and Body Characteristics Monitoring

This section presents the predictive models for identifying the behaviors and body characteristics of *Gallus domesticus* and predicting the possible diseases that they might have.

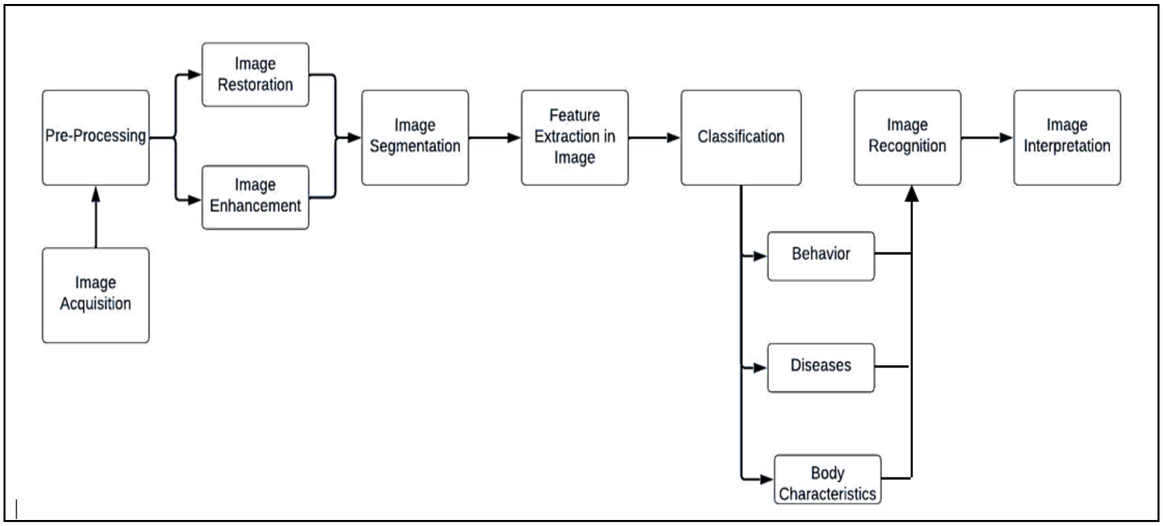


Figure 3.13 Image Processing Block Diagram

Figure 3.13 shows the concept of imaging processing starting with acquiring the image from the camera. The acquired image would then be pre-processed in which the image would be restored and enhanced. Image Segmentation is the next step in which it divides the digital image into several image segments and extraction of features in the image. For classifying the current state of *Gallus domesticus*, the use of a predictive model would be implemented. The activities of *Gallus domesticus* would serve as patterns to identify their current behavior and body characteristics, as well as predicting the possible disease they might have. Lastly, the acquired image would be recognized and interpreted by the system.

The system would implement Artificial Neural Networks (ANN) for predicting the behavior and body characteristics of *Gallus* *domesticus*. The activities of *Gallus domesticus* would serve as the patterns for determining their current condition.

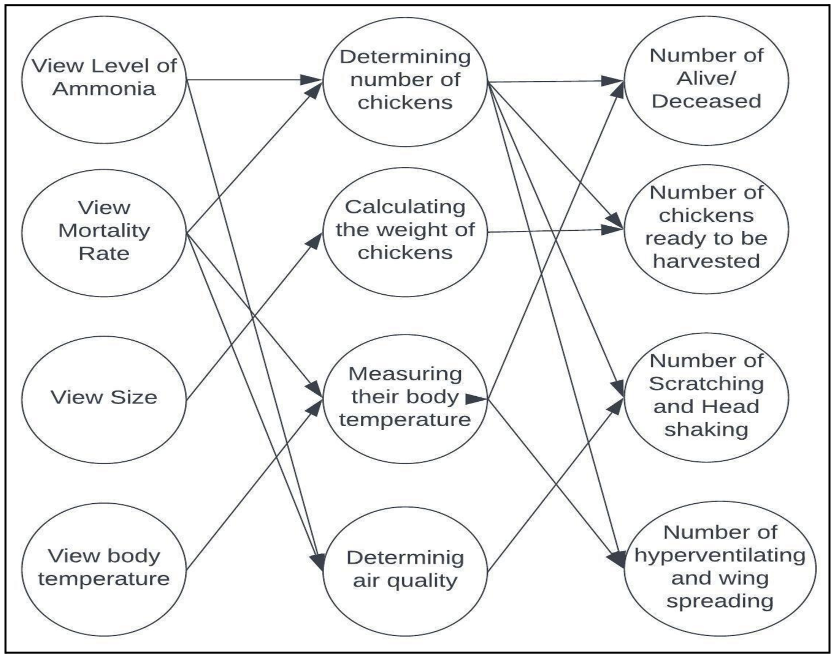


Figure 3.14 Artificial Neural Networks Model for Behaviors

Figure 3.14 shows the patterns of what unusual behaviors a certain *Gallus domesticus* might experience in the coop. These include the following:

1. Hyperventilating

- Ventilating for *Gallus domesticus* is normal, however, it is a sign of a hyper-ventilating behavior if they frequently show beak gaping.
- There are internal problems present, such as lung diseases.

2. Wing Spreading

- Behaviors such as flapping and spreading their wings are frequently shown.
- The certain *Gallus domesticus* might be experiencing heat-related stress and diseases.

3. Head Scratching

- The fowls frequently scratch their head and neck.
- There are external problems such as wounds, lice, and ticks

4. Head Shaking

- The fowls frequently shake their heads.
- The fowls might have ticks and lice.

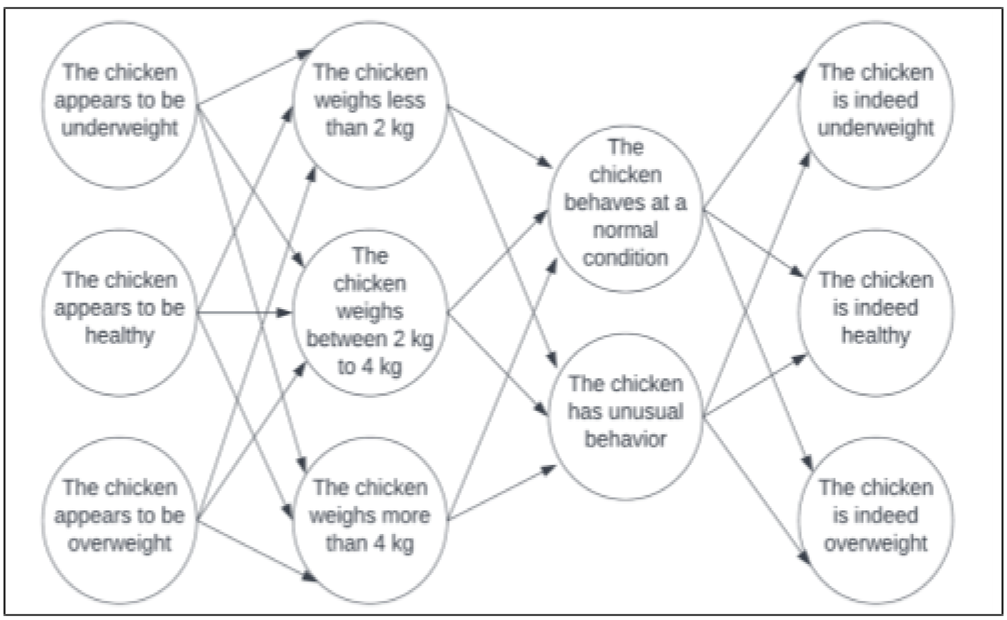


Figure 3.15 Artificial Neural Networks Model for Body Characteristics

Figure 3.15 shows the patterns to determine the body characteristics of *Gallus domesticus*. The average weight range of *Gallus domesticus* is estimated at 2.6 kg to 4 kg. A significant increase and decrease in weight might be a potential risk. The fowls weighing 1 kg less and 1 kg more than the average weight are subject to further monitoring and veterinarian care. In addition, the physical appearance of *Gallus domesticus* would also be used to identify the current condition of the coop.

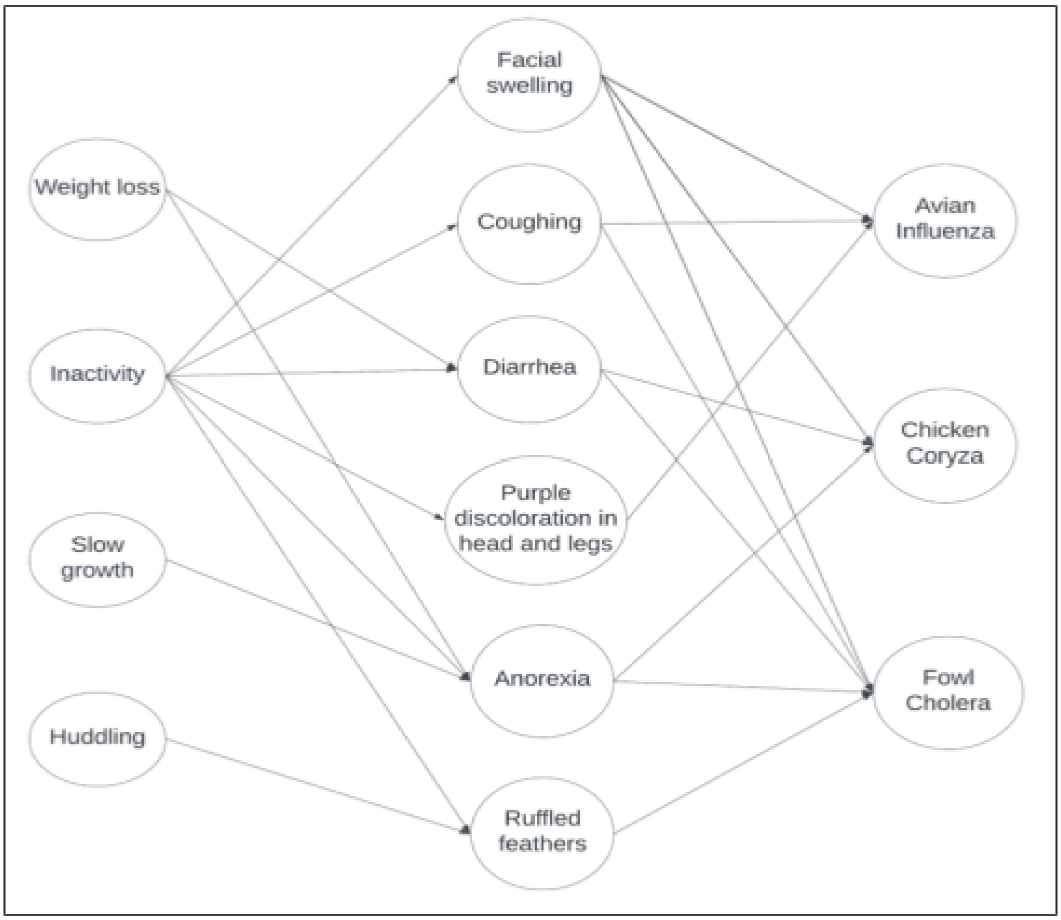


Figure 3.16 Artificial Neural Networks Model for Possible Diseases

Figure 3.16 shows the possible diseases *Gallus domesticus* might experience in the coop, with the corresponding symptoms as patterns:

1. Avian Influenza

- Symptoms of Avian Influenza may be noticed through facial sweating, nasal discharge, sneezing, coughing, purple discoloration of the head and legs, and reduced egg production.

2. Chicken Coryza
   - Symptoms of Chicken Coryza may be noticed through facial sweating, nasal discharge, sneezing, diarrhea, and anorexia.
3. Fowl Cholera
   - Symptoms of Fowl Cholera may be noticed through facial sweating, nasal discharge, coughing, diarrhea, anorexia, swollen foot and joint, lameness, ruffled feathers, and reduced egg production.

3.5.2 Web Application Development of Real-Time Monitoring System of Body Characteristics and Behavior of *Gallus domesticus*

This section presents the flowchart for the website of the prototype which discusses the flow of operation of the website as a whole.

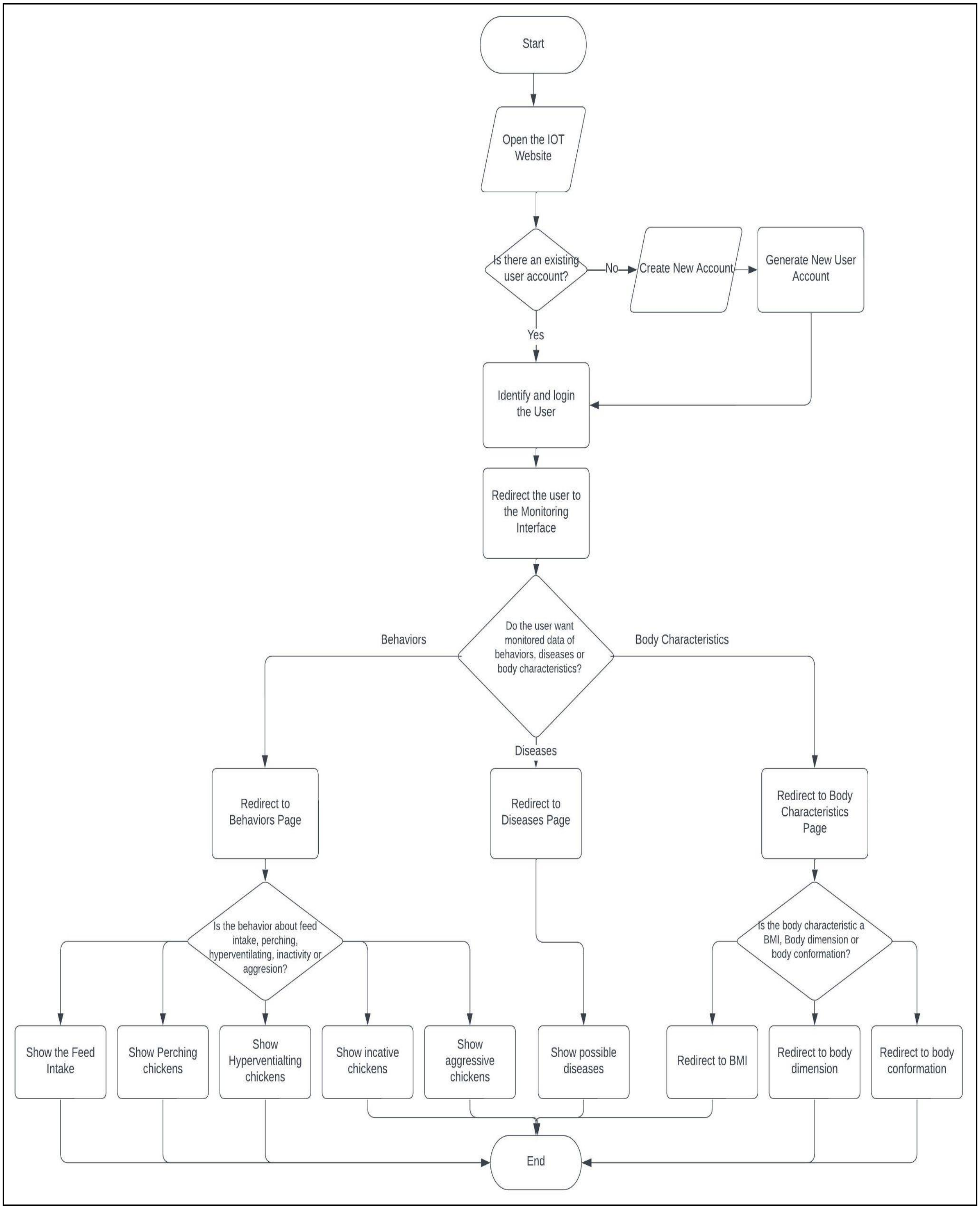


Figure 3.17 Website Flowchart Diagram

Figure 3.17 shows the flow chart diagram for the website. At first, the user opens the web application and is redirected to the registration page of the website wherein they will input some personal information. After the user has inputted their personal information, the system will identify the user and its own respective ID of the camera. If the account is not existing, the website will prompt the user to create a new account on the registration page. After successfully logging in, the system will ask for the camera ID of the user and will be redirected to the user page where they can view the body characteristics, behaviors, possible diseases and manage their account.

3.6 User Acceptance Testing

The study will be using Technology Acceptance Model (TAM) which primarily focuses on the effectiveness of the prototype's technology. The parameters will be rated according to its effectiveness via rating scale. The dimensions (height and weight) of the fowls and their body mass index (BMI) will be measured using the actual measurement gathering and computer vision. It also guarantees that the prototype is capable of handling real-world activities and meeting development standards.

3.7 Statistical Analysis

The results that would be obtained between the manual measurement and estimated measurement of weight of the chickens would be evaluated with the use of t-test: Paired Two sample of Means.

$$t = \frac{\bar{d}}{s/\sqrt{n}}$$

where: d = difference of means

s^2 = sample variance

n = sample size

t = the calculated difference

The efficiency of the camera would be calculated with the use of the percentage error formula, where the number of sick chickens monitored by the camera would be compared to the number of sick chickens manually checked.

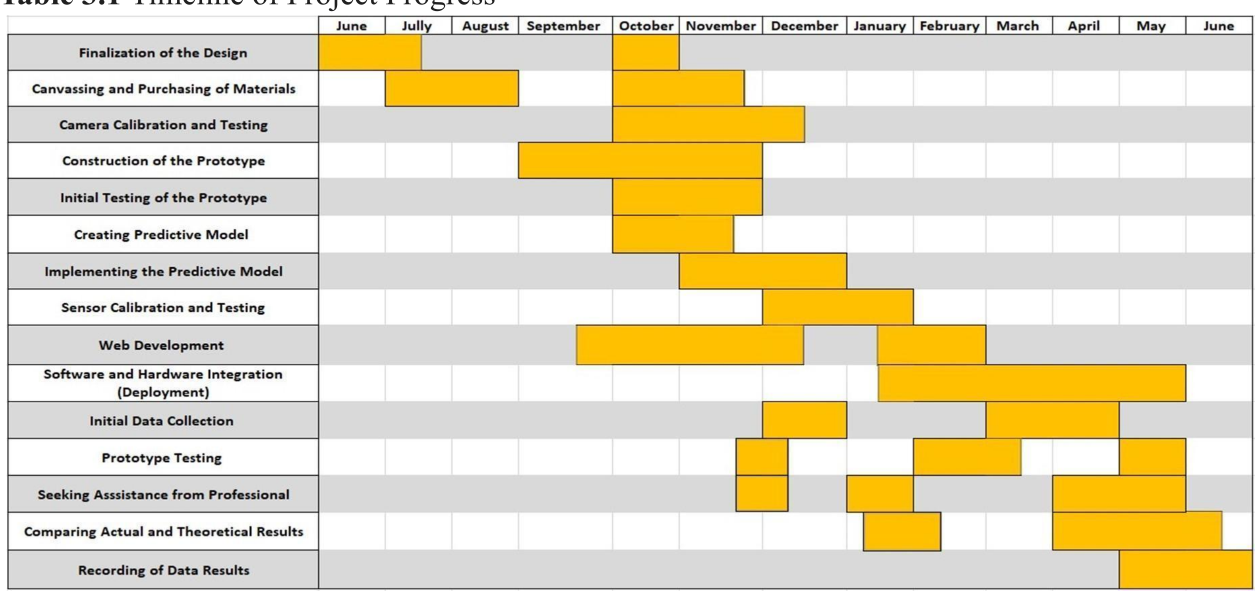
$$\%Error = \frac{|c - m|}{m} \times 100$$

where: c = number of chickens with unideal condition checked by the camera

m = number of chickens with unideal condition manually checked

3.8 Project Workplan

Table 3.1 Timeline of Project Progress



CHAPTER 4

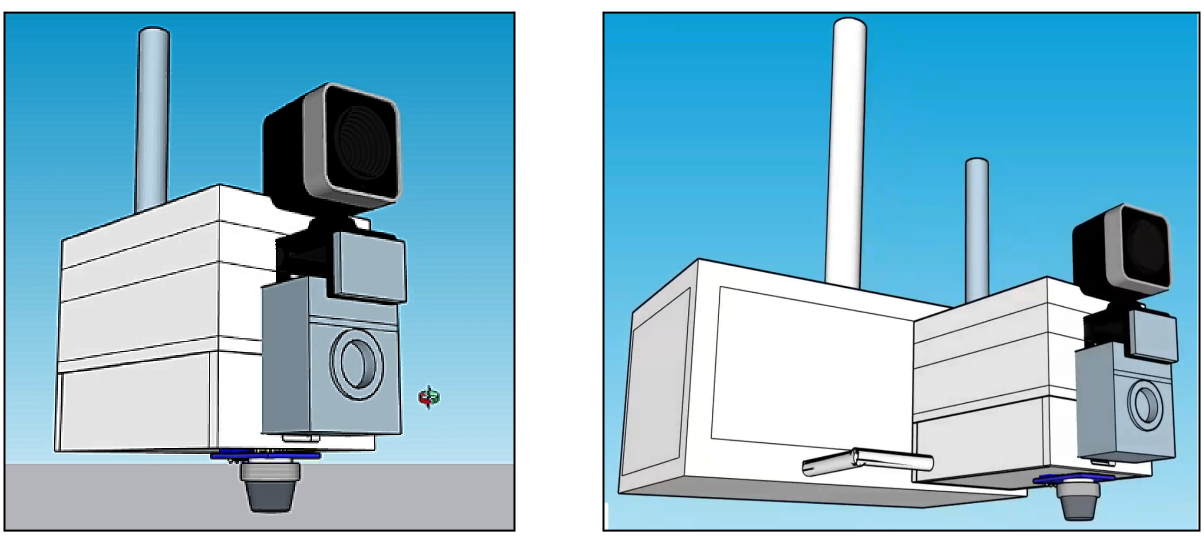
DATA AND RESULTS

This chapter contains the project technical description, project structural description, and tabulation and analysis of results relative to the tests conducted.

4.1 Project Technical Description

The technology developed for project entitled "GalluSpy Cam: Real-Time Monitoring System of Body Characteristics and Behavior of *Gallus Domesticus* using Computer Vision via Internet-of-Things (IoT)" aims to determine the body characteristics and behavior of the chicken, specifically the Silkie Chicken and 45 days Chicken using Microsoft Webcam and MLX90640 as thermal technology for the room temperature. All the data collected will be processed through Raspberry Pi 4 and can be accessed using a web application.

4.2 Project Structural Design

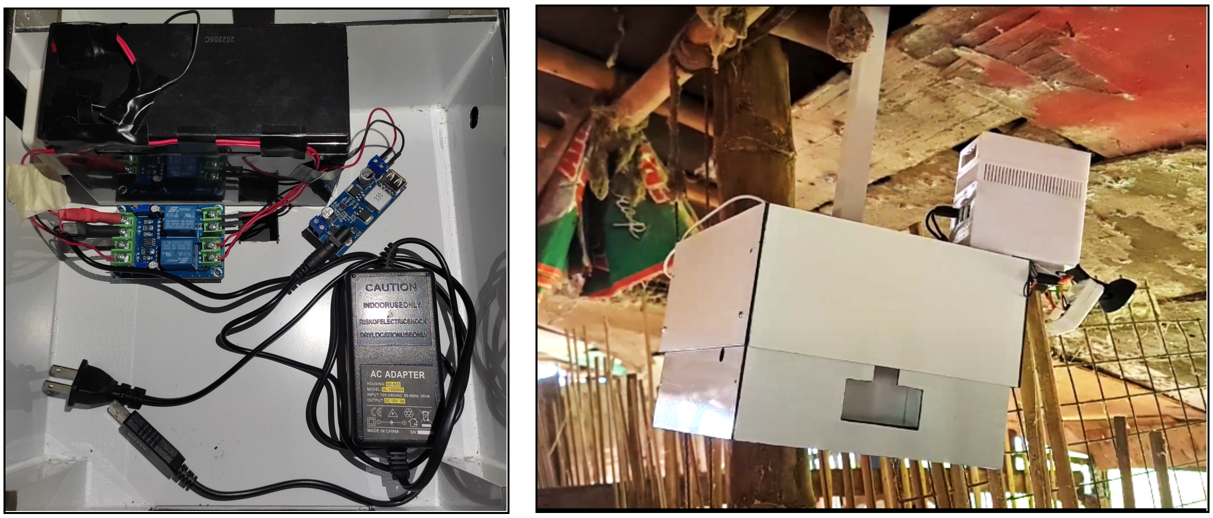


(a) (b)

Figure 4.1 GalluSpy Cam 3D Prototype: (a) Camera and sensor enclosure only and (b) Camera and sensor enclosure with power supply

Figure 4.2 Galluspy Cam Prototype: (a) Front view (b)Left-side view (c) Chassis

Figure 4.2 shows the design of the prototype casing which is primarily made of aluminum. The aluminum casing has a length of 12 inches, height of 16 inches, and width of 8 inches. In addition, it can withstand approximately 10 kg of weight.



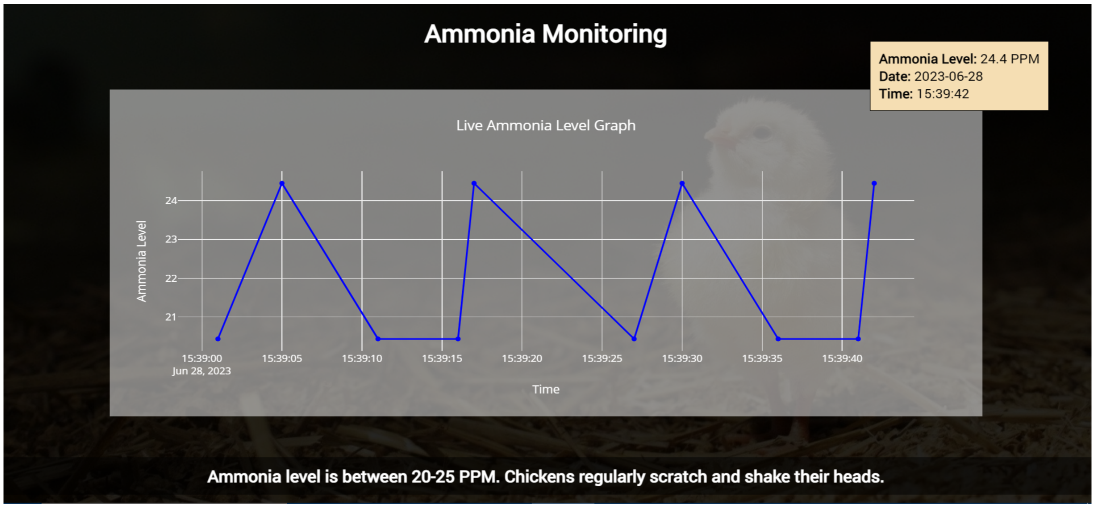
(a) (b)

Figure 4.3 Actual placement of the components: (a) Power supply components inside the chassis (b)Prototype deployed in the site

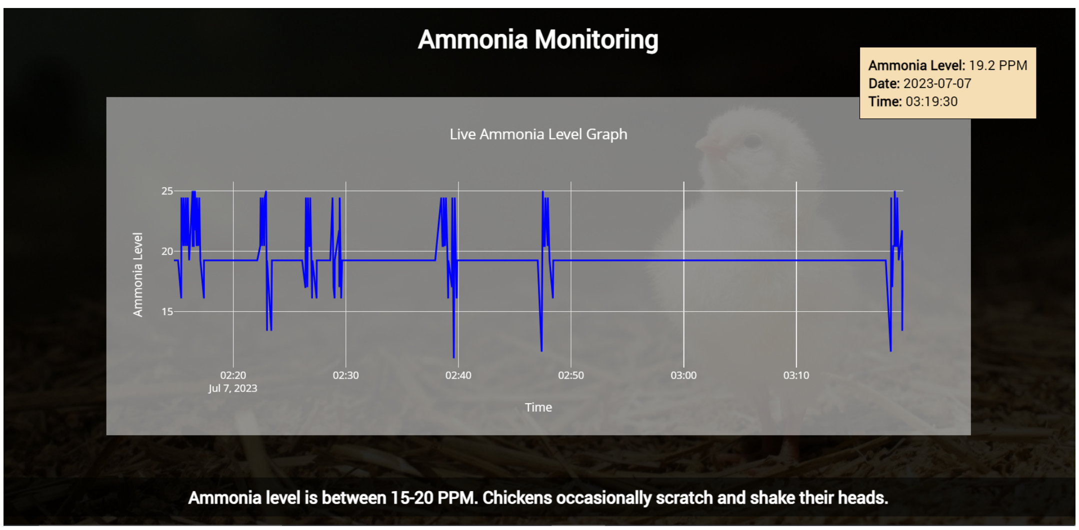
Figure 4.3 shows the positioning and arrangement of every component used in the project. The prototype is divided by two sections, the aluminum case has the components for power supply which consist of battery, buck converter, relay module, and AC adapter. The other one has the main components like the raspberry pi, thermal camera, air quality sensor, and Microsoft lifecam.

The camera was positioned at the left top corner of the coop, which was 1.5 to 2 meters from the base of the chickens. According to the datasheet of the camera and sensors, and by conducting several tests, the corner was determined as the suitable place to put the system on.

4.3 Testing Results



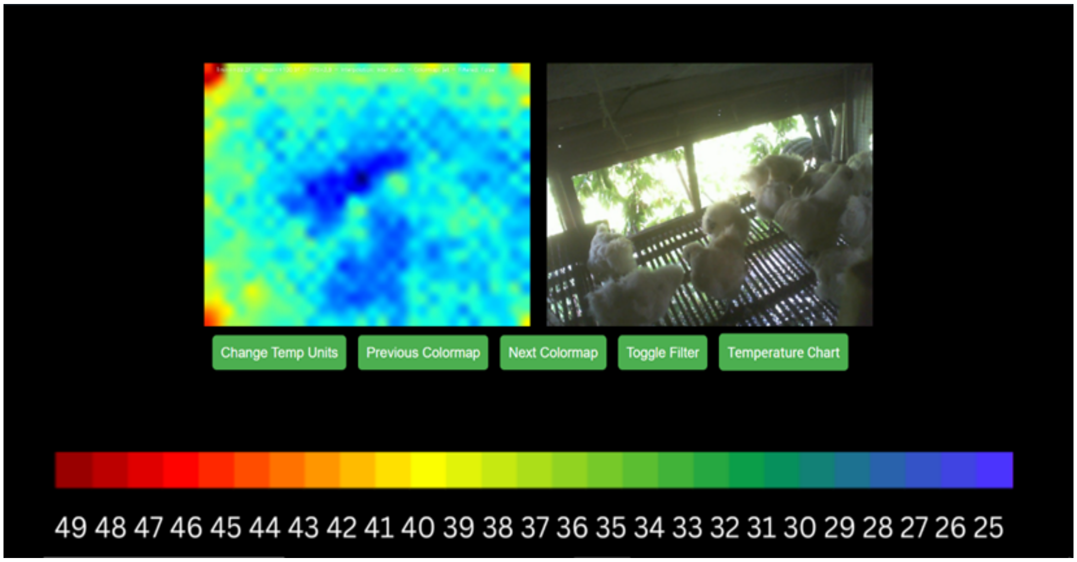
(a)



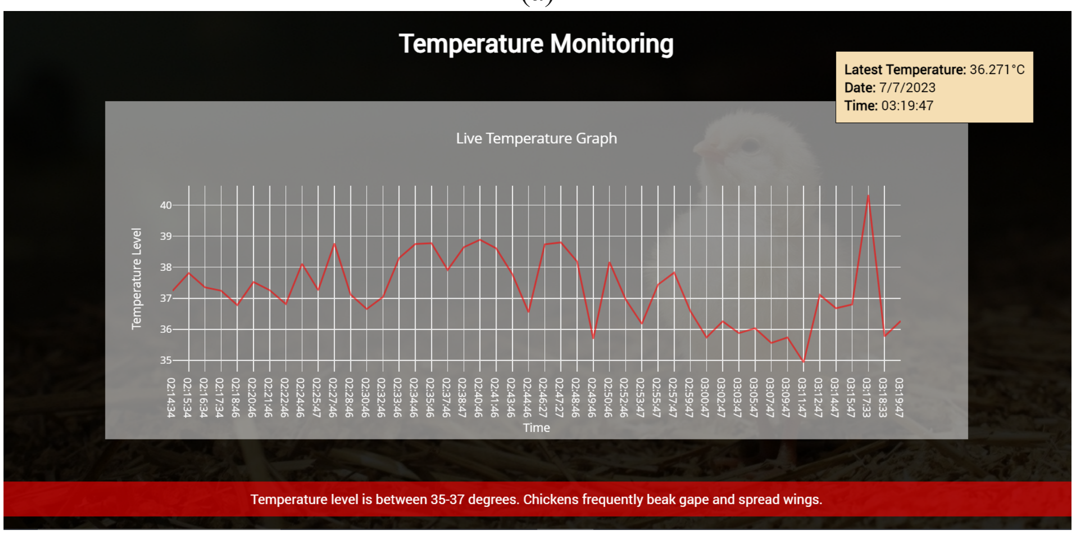
(b)

Figure 4.4 Graph of Ammonia Levels: (a) Ammonia level showing various peak (b) Ammonia level showing stability with sudden peak

Figure 4.4 shows the graph of ammonia levels inside the coop. There are spikes or changes in the graph indicating the level of ammonia present and being detected. At the bottom of the graph there is a note regarding the behavior of the chickens under certain ammonia levels.



(a)



(b)

Figure 4.5 Temperature Monitoring: (a)Live Temperature Footage (b)Live Temperature Graph

Figure 4.5 shows the live footage of the thermal camera and Microsoft lifecam inside the coop. It also shows the graphical representation of the temperature detected. At the bottom of the graph is a notification about the behaviors of the chickens at a certain temperature.

4.4 Project Evaluation

The system is deployed and monitored for five days with intervals between morning until evening. The functionality of the system was evaluated through daily data gathering.

4.4.1 Ammonia Monitoring System

Table 4.1 Level of Ammonia and Mortality Rate per Day (AM/PM)

Day	Number of Chickens	AM/PM	Weight	Deceased	Level of Ammonia (PPM)
Day 1	19	5 hours	1.6Kgs	None	25 PPM
	19	5 hours	1.6Kgs	None	26 PPM
Day 2	19	5 hours	1.6Kgs	None	25 PPM
	19	5hours	1.6Kgs	None	24 PPM
Day 3	19	5 hours	1.6Kgs	None	23 PPM
	19	5 hours	1.6Kgs	None	24 PPM
Day 4	19	5 hours	1.6Kgs	None	20 PPM
	19	5 hours	1.6Kgs	None	20 PPM
Day 5	19	5 hours	1.6Kgs	None	24 PPM
	19	5 hours	1.6Kgs	None	20 PPM

Table 4.1 summarizes the key data regarding chicken and ammonia levels observed over the five-day period. The table provides details such as the number of chickens, time of observation, chicken weight, mortality, and the recorded levels of ammonia in parts per million (PPM) for each day.

4.4.1.1 Level of Ammonia per Day (AM/PM)

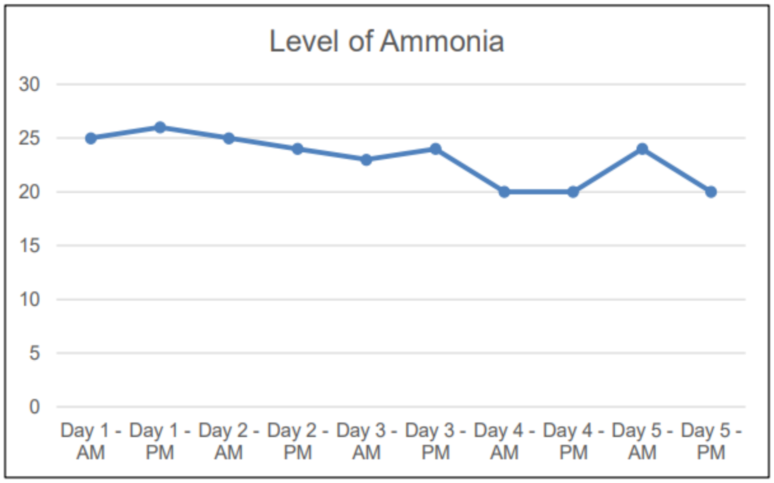


Figure 4.6 Level of Ammonia per Day (AM/PM)

Figure 4.6 shows the ammonia level of the chicken coop alerts the user and shows behavioral frequency level whenever the reading goes above a certain ammonia level on the same day as it is measured. The ideal ammonia level of chicken coop is 20 to 25 PPM.

4.4.1.2 Behavior and Level of Frequency According to the Level of Ammonia

Table 4.2 Behavior and Level of Frequency of Ammonia per Day (AM/PM)

Ammonia Level	Chickens Scratching	Scratching - Frequency	Chickens Head Shaking	Head Shaking - Frequency
20 PPM	4	6	12	13
20 PPM	3	5	13	9
20 PPM	5	3	10	10
23 PPM	4	4	15	20
24 PPM	3 to 7	4	17	21
24 PPM	5	5	16	21
24 PPM	5	5	12	15
25 PPM	5	5	15	19
25 PPM	4 to 6	5	14	20
26 PPM	5 to 6	4	16	21

The data in Table 4.2 indicates the number of chickens observed exhibiting scratching and head shaking behaviors at different levels of ammonia concentration. The frequency of each behavior is also recorded. In this dataset, the researchers observed that higher ammonia levels (23 PPM to 26 PPM) correspond to increased occurrences of scratching and head shaking behaviors. Specifically, at ammonia levels of 23 PPM, 24 PPM, 25 PPM, and 26 PPM, the number of chickens scratching and head shaking increases compared to the observations at 20 PPM ammonia level.

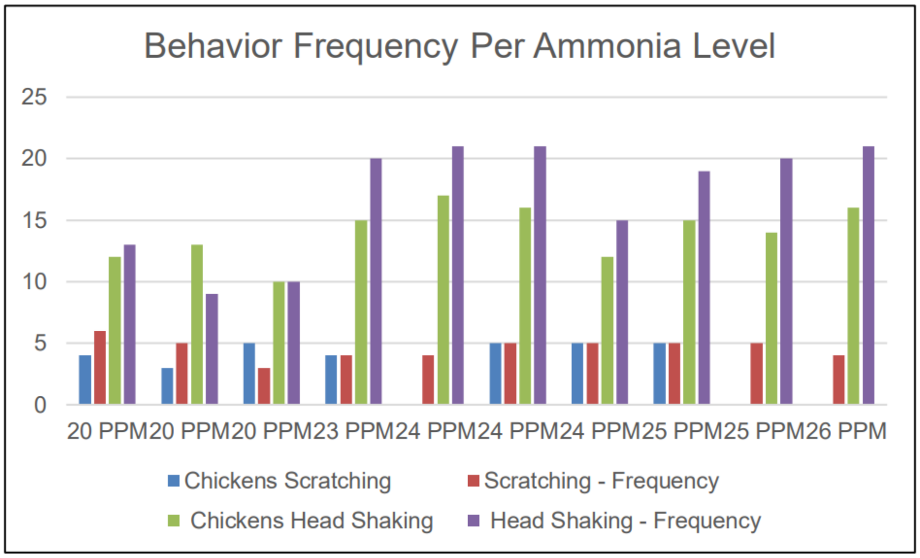


Figure 4.7 Behavior and Level of Frequency of Ammonia per Day (AM/PM)

Figure 4.7 shows the amount of fowls showing specific behaviors in response to the ammonia level inside the coop, alongside with the frequency of these behaviors. Such behaviors include scratching and head shaking which are signs of irritation caused by increasing ammonia levels. For each ammonia level, you can plot four bars side by side: two for the number of chickens scratching and two for the number of chicken's head shaking. The height of each bar represents the corresponding count or frequency. The fowls were frequently observed to show the behaviors as the ammonia level inside the coop increases.

4.4.2 Behavior and Level of Frequency According to the Temperature

4.4.2.1 Level of Temperature per Day (AM/PM)

Table 4.3 Level Temperature and Mortality Rate per Day (AM/PM)

Day	AM/PM	Weight	Number of Chickens	Level of Temperature (Celsius)
Day 1	5 hours	1.6 Kgs	0	40 - 37
	5 hours	1.6 Kgs	0	39-35
Day 2	5 hours	1.6 Kgs	0	42-38
	5 hours	1.6 Kgs	0	41-39
Day 3	5 hours	1.6 Kgs	0	38 - 36
	5 hours	1.6 Kgs	0	39-37
Day 4	5 hours	1.6 Kgs	0	34 - 32
	5 hours	1.6 Kgs	0	35-31
Day 5	5 hours	1.6 Kgs	0	35-33
	5 hours	1.6 Kgs	0	34-33

Table 4.3 displays the recorded temperature levels over a period of five days. The temperatures are measured in degrees Celsius and each entry in the table represents a specific time point within a given day. The ideal temperature level of chickens is 37 °C to 40 °C. The temperature on Day 4 indicates relatively cooler conditions than the other days due to the occurrence of a typhoon.

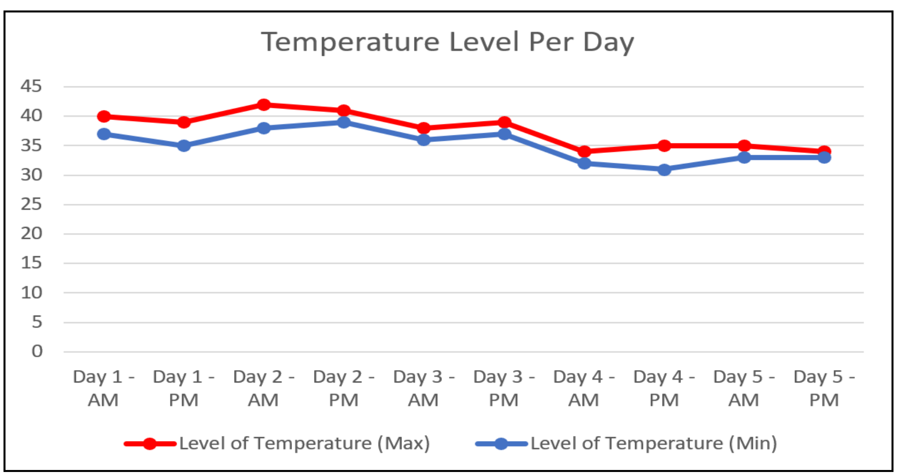


Figure 4.8 Level Temperature per Day (AM/PM)

Figure 4.8 shows the temperature level of the chickens, which alerts the user and shows behavioral frequency level whenever the reading goes above a certain temperature level on the same day as it is measured. The ideal temperature level of chicken coop is 37 °C to 40 °C.

4.4.2.2 Behavior and Level of Frequency According to the Temperature

Level of Temperature (Celsius)	Beak Gaping-No. of Chickens	Beak Gaping-Frequency	Wing Spreading-No. of Chickens	Wing Spreading - Frequency
34 - 32	1	Never	0	0
34-33	0	Never	1	1
35-31	0	Never	0	1
35-33	0	Never	1	1
38 - 36	4	Sometimes	4	2
39-35	10	Always	4 to 5	3
39-37	5	Sometimes	4	3
40 - 37	12	Always	3 to 4	3
41-39	13	Always	5	3
42-38	11	Always	5	4

Table 4.4 Behavior and Frequency Level of Temperature per Day (AM/PM)

The Table 4.4 demonstrates how the behaviors of beak gaping and wing spreading are associated with different temperature ranges. The researchers observed that these behaviors were more prevalent at higher temperatures, suggesting that they are linked to thermal regulation in chickens. The highest number of chickens (13) displaying beak gaping was recorded at a temperature range of 39 °C to 41 °C. The highest number of chickens (5) displaying wing spreading was observed at temperature ranges of 39 °C to 41 °C and 38 °C to 42 °C.

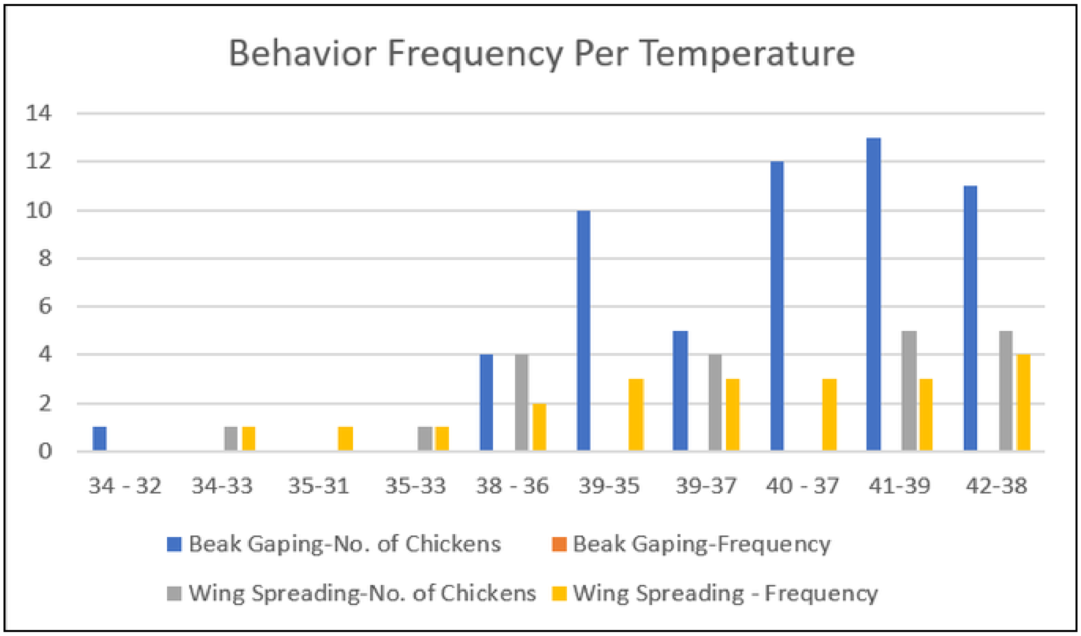


Figure 4.9 Behavior and Frequency Level of Temperature per Day (AM/PM)

In figure 4.9 the x-axis of the chart represents the different temperature ranges in degrees Celsius, ranging from the lowest range to the highest range observed in the data. The y-axis, on the other hand, represents the number of chickens exhibiting beak gaping and wing spreading behaviors, as well as the frequency of these behaviors. Four bars are plotted side by side: two for the number of chickens exhibiting beak gaping and two for the number of chickens exhibiting wing spreading. The height of each bar represents the corresponding count or frequency.

4.4.3 Weight Prediction Results and Discussion

Table 4.5 Data for Predicted and Measured Weight

Day	AM/PM	Predicted	Measured	Difference	Error
Day 1	5 Hours	1.6	1.6	0	0%
	5 Hours	1.6	1.6	0	0%
Day 2	5 Hours	1.6	1.7	0.1	5.88%
	5 Hours	1.6	1.7	0.1	5.88%
Day 3	5 Hours	1.7	1.7	0	0%
	5 Hours	1.7	1.8	0.1	5.88%
Day 4	5 Hours	1.8	1.8	0	0%
	5 Hours	1.8	1.9	0.1	5.88%
Day 5	5 Hours	1.9	1.9	0	0%
	5 Hours	1.9	1.9	0	0%
			Mean Percent Error		2.352 %

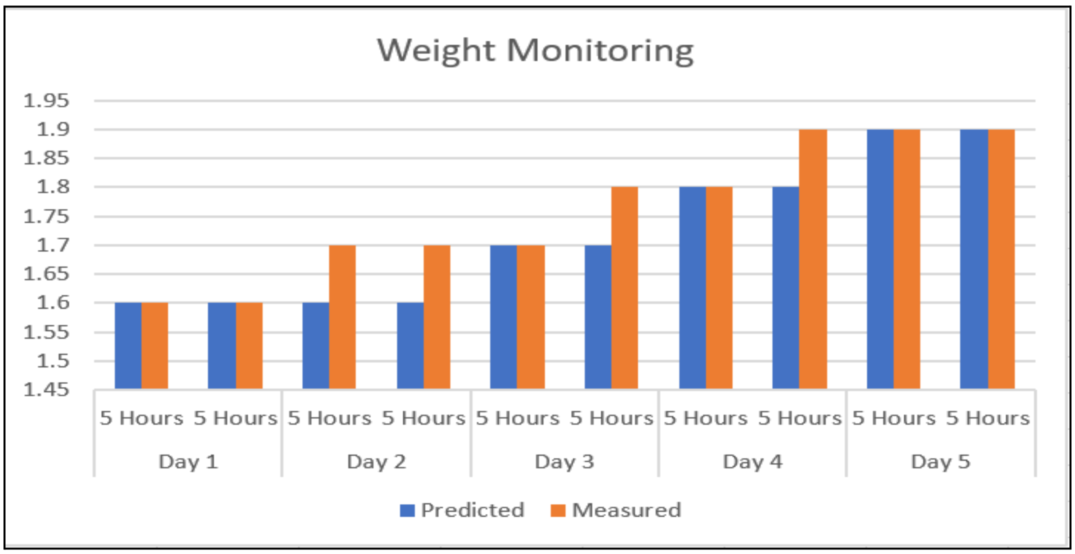


Figure 4.10 Data for Predicted and Measured Weight

Table 4.5 and Figure 4.10 test the accuracy of weight prediction; a comparison was created between the actual and the predicted weight using the Galluspy Cam camera. The table shows the recorded weights of chickens in five days between intervals morning and afternoon. With the data assimilated, it was shown that the mean percentage error of actual and predicted is 2.352%.

4.5 Statistical Analysis

The researchers sampled 10 chickens and weighed both manually and using the system. By using paired t-tests, the two-tailed p-value was 0.00062064. Since the p-value was smaller than the significance level, which is 0.05, the null hypothesis was rejected. The null hypothesis assumed that there was no significant difference between the weights obtained through manual weighing and weighing using the system. However, the low p-value suggests that this assumption was unlikely to be true.

Therefore, it could be concluded that there was strong evidence to support the alternative hypothesis, which stated that there would be a significant difference between the two weighing methods. In this case, weighing using the system demonstrates a statistically significant difference in measuring chicken weights compared to manual weighing.

Table 4.6 Manual and Predicted Weight

Chickens	Monitored (kg)	Manual (kg)
1	1.6	2
2	1.6	2
3	1.6	1.9
4	1.6	1.8
5	1.6	1.6
6	1.6	2
7	1.6	1.8
8	1.6	1.7
9	1.6	1.8
10	1.6	1.7

Paired T-test
0.00062064

CHAPTER 5

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

This chapter contains the summary of findings, conclusions, and recommendations for further improvement and development of the study.

5.1 Summary of Findings

This research study aimed to develop and evaluate a comprehensive monitoring system for chicken coops, utilizing air quality sensors, camera technology, and thermal imaging integrated through computer vision. The objectives were to implement a predictive model, create an IoT-based interface, and assess the system's performance.

The developed monitoring system, equipped with air quality sensors and thermal imaging, successfully captured and analyzed these data points. The predictive model integrated into the system demonstrated accuracy in determining chicken behaviors and conditions based on the collected data, enabling prompt interventions to address potential health issues caused by ammonia levels and temperature variations.

The IoT-based interface provided convenient access to the monitoring system's data, enabling users to analyze information regarding ammonia levels and temperature fluctuations. This real-time monitoring capability empowers poultry farmers to make informed decisions and implement preventive measures to mitigate potential risks associated with suboptimal environmental conditions.

The frequency levels of ammonia and temperature deviations were also examined. The data indicated that the frequency of ammonia levels exceeding the recommended threshold was observed in certain instances, suggesting a potential concern for the health and well-being of the chickens. Similarly, the frequency of temperature fluctuations beyond the optimal range was identified, indicating the need for timely interventions to maintain a suitable environment for the chickens.

The results of the study revealed varying levels of ammonia and temperature within the chicken coops. The ammonia levels ranged from 20 PPM to 26 PPM during the observation period, indicating the presence of ammonia gas in the air. The temperature data showed fluctuations between 32°C and 42°C over a span of five days, reflecting the changing environmental conditions within the coops.

Regarding temperature, an increase in temperature corresponded to an increased frequency of beak gaping and wing spreading. This suggests that the chickens exhibit these behaviors more often as the temperature rises. It indicates that high temperatures may cause discomfort or stress to the chickens, leading to such behaviors.

In terms of ammonia levels, higher concentrations were associated with an increased frequency of scratching and head shaking. These behaviors are likely responses to the presence of ammonia, which can irritate the chickens' respiratory system and overall well-being. The elevated frequencies of scratching and head shaking serve as indicators of potential health issues caused by ammonia levels.

5.2 Conclusion

The following inferences are drawn from the collected data, results of the tests performed, and analysis of the results:

1. The researchers successfully developed a monitoring system integrating the usage of air quality sensors, cameras, and thermal imaging to monitor

the chicken's environment through calibration, python programming and website development.

2. The researchers successfully implemented the use of ANN predictive model which predicted the behavior of the fowls. This predictive model was successful in processing the footage that was captured by the camera, which then displayed the predicted behavior on the web application interface.
3. The researchers also successfully created an IoT-based using python flask and HTML interface providing convenient and interactive access to the data, enabling poultry farmers to monitor the behaviors and conditions of their chickens through web application and is also able to download history logs/data via excel or csv file .
4. The researchers successfully highlighted the system's accuracy compared to the manual measurement, which has 2.352 % error, in determining behaviors, detecting environmental variations, and identifying potential health risks by comparing it to manual health conditioning and measurements.

5.3 Recommendation

For further improvement of the study, the researchers recommend to:

1. Include an additional type of chicken in future studies. By incorporating a diverse range of chicken breeds or varieties, a more comprehensive understanding of various factors, such as behavior, growth, and health, can be achieved.

2. Use PTZ (Pan-Tilt-Zoom) cameras which are capable of remote control over pan, tilt, and zoom functions. These cameras could provide broader coverage and the ability to focus on specific details.
3. Use multiple cameras in order to compensate for the blind spots of each camera. With the use of multiple cameras, variability could be introduced in the captured images, and it would also give a more comprehensive view for the machine learning model to be more accurate.
4. Gather a larger dataset of chicken images to enhance the training process. This could involve capturing images of chickens from different angles, under various lighting conditions, and with diverse backgrounds. Increasing the dataset size with high-quality images could help the model learn more robust representations of chicken features and improve its accuracy.

REFERENCES

[1] Natalie Berkhout, “Philippines: Chicken overtakes pork as preferred protein,” Poult. WORLD, 2020.

[2] H. M. Hafez and Y. A. Attia, “Challenges to the Poultry Industry: Current Perspectives and StrategicFuture After the COVID-19 Outbreak,” Front. Vet. Sci., vol. 7, no. August, 2020, doi: 10.3389/fvets.2020.00516.

[3] Annsofie Wahlstrom, “The importance of 7-day weight,” Poult. World, 2013.

[4] Growel Agrovet Private Limited, “Importance of Seven Days Poultry Chicks Weight,” Slideshare, p. 2,2015.

[5] A. Pezzuolo, V. Milani, D. H. Zhu, H. Guo, S. Guercini, and F. Marinello, “On-barn pig weight estimation based on body measurements by structure-from-motion (SfM),” Sensors (Switzerland), vol. 18, no. 11, pp. 1–12, 2018, doi: 10.3390/s18113603.

[6] I. Nyalala, C. Okinda, C. Kunjie, T. Korohou, L. Nyalala, and Q. Chao, “Weight and volume estimation of poultry and products based on computer vision systems: a review,” Poult. Sci., vol. 100, no. 5, p. 101072, 2021, doi: 10.1016/j.psj.2021.101072.

[7] B. Bourgeois et al., “Clinically applicable optical imaging technology for body size and shape analysis: Comparison of systems differing in design,” Eur. J. Clin. Nutr., vol. 71, no. 11, pp. 1329–1335, 2017, doi:10.1038/ejcn.2017.142.

[8] I. Nyalala, C. Okinda, C. Kunjie, T. Korohou, L. Nyalala, and Q. Chao, “Weight and volume estimation of poultry and products based on computer vision systems: a

review," Poult. Sci., vol. 100, no. 5, p. 101072, 2021, doi: 10.1016/j.psj.2021.101072.

[9] F. Marinello, A. Pezzuolo, D. Cillis, F. Gasparini, and L. Sartori, "Application of Kinect-Sensor for three-dimensional body measurements of cows," Precis. Livest. Farming 2015 - Pap. Present. 7th Eur. Conf. Precis. Livest. Farming, ECPLF 2015, no. January 2016, pp. 661–669, 2015.

[10] F. Marinello, A. Pezzuolo, D. Cillis, F. Gasparini, and L. Sartori, "Application of Kinect-Sensor for three-dimensional body measurements of cows," Precis. Livest. Farming 2015 - Pap. Present. 7th Eur. Conf. Precis. Livest. Farming, ECPLF 2015, no. January 2016, pp. 661–669, 2015.

[11] A. Pezzuolo, V. Milani, D. H. Zhu, H. Guo, S. Guercini, and F. Marinello, "On-barn pig weight estimation based on body measurements by structure-from-motion (SfM)," Sensors (Switzerland), vol. 18, no. 11, pp. 1–12, 2018, doi: 10.3390/s18113603.

[12] G. Lowe et al., "Automated collection and analysis of infrared thermograms for measuring eye and cheek temperatures in calves," Animals, vol. 10, no. 2, 2020, doi: 10.3390/ani10020292.

[13] K. Hill, "5 important reasons why you must weigh your livestock regularly | 2018-09-04 | The NationalProvisioner," Industry News, p. 1, 2018. [Online]. Available: https://www.provisioneronline.com/articles/106701-important-reasons-why-you-must-weigh-your-livestock -regularly

[14] K. Wurtz et al., Recording behaviour of indoor-housed farm animals automatically using machine vision technology: A systematic review, vol. 14, no. 12. 2019. doi: 10.1371/journal.pone.0226669.

[15] N. Li et al., “Review: Automated techniques for monitoring the behaviour and welfare of broilers and laying hens: towards the goal of precision livestock farming,” Animal, vol. 14, no. 2, pp. 617–625, 2020, doi: 10.1017/S1751731119002155.

[16] J. Y. Noh et al., “Thermal Image Scanning for the Early Detection of Fever Induced by HighlyPathogenic Avian Influenza Virus Infection in Chickens and Ducks and Its Application in Farms,” Front. Vet. Sci., vol. 8, no. May, pp. 1–8, 2021, doi: 10.3389/fvets.2021.616755.

[17] Z. Wang, S. Shadpour, E. Chan, V. Rotondo, K. M. Wood, and D. Tulpan, “ASAS-NANP SYMPOSIUM: Applications of machine learning for livestock body weight prediction from digital images,” J. Anim. Sci., vol. 99, no. 2, pp. 1–15, 2021, doi: 10.1093/jas/skab022.

[18] R. Dohmen, C. Catal, and Q. Liu, “Computer vision-based weight estimation of livestock: a systematic literature review,” New Zeal. J. Agric. Res., vol. 65, no. 2–3, pp. 227–247, 2022, doi: 10.1080/00288233.2021.1876107.

[19] P. Li, H. Lu, F. Wang, S. Zhao, and N. Wang, “Detection of sick laying hens by infrared thermal imaging and deep learning,” J. Phys. Conf. Ser., vol. 2025, no. 1, 2021, doi: 10.1088/1742-6596/2025/1/012008.

[20] S. Loker, “Mass customization and sizing,” *Sizing in Clothing*, pp. 246–263, 2007, doi:10.1533/9781845692582.246. [Online]. Available: https://www.sciencedirect.com/topics/engineering/body-measurement.

[21] "BMI (Body Mass Index): What It Is & How To Calculate," *Cleveland Clinic*, 2022. [Online]. Available: https://my.clevelandclinic.org/health/articles/9464-body-mass-index-bmi.

[22] "Estimation of live body weight using zoometrical measurements for improved marketing of indigenous chicken in the Lake Victoria basin of Uganda," *Cipav.org.co*, 2022. [Online]. Available: https://www.lrrd.cipav.org.co/lrrd23/8/sema23170.htm.

[23] M. Mendeş, E. Dinçer, and E. Arslan, "Profile Analysis and Growth Curve for Body Mass Index ofBroiler Chickens Reared Under Different Feed Restrictions in Early Age," *Archives Animal Breeding*, vol. 50, no. 4, pp. 403–411, Oct. 2007, doi: 10.5194/aab-50-403-2007. [Online]. Available: https://aab.copernicus.org/articles/50/403/2007/.

[24] W. Ma, Q. Li, J. Li, L. Ding, and Q. Yu, "A method for weighing broiler chickens using improved amplitude-limiting filtering algorithm and BP neural networks," *Information Processing in Agriculture*, vol. 8, no. 2, pp. 299–309, Jun. 2021, doi: 10.1016/j.inpa.2020.07.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214317320301888.

[25] L. Huang *et al.*, "Body Dimension Measurements of Qinchuan Cattle with Transfer Learning from LiDAR Sensing," *Sensors*, vol. 19, no. 22, p. 5046, Nov. 2019, doi: 10.3390/s19225046. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6891291/.

[26] "chicken | bird | Britannica," *Encyclopædia Britannica*. 2022 [Online]. Available: https://www.britannica.com/animal/chicken.

[27] N. P. Petrus, K Kangootui, E. Kandiwa, and Borden Mushonga, "Relationship of Age and Live weight to Linear Body Traits in Female Intensively Reared Boschveld Chicken...," *ResearchGate*, Jul. 23, 2019. [Online]. Available: https://www.researchgate.net/publication/334625469_Relationship_of_Age_and_Live_weight_to_Linear_ Body_Traits_in_Female_Intensively_Reared_Boschveld_Chicken_in_Namibia.

[28] Clement Gyeabour Kyere, Sadick Muumin, G. Aryee, and P. Atta, "Relationship between body weight and linear body measurements in the Cobb broiler chicken," *ResearchGate*, Nov. 23, 2020. [Online]. Available: https://www.researchgate.net/publication/346106367_Relationship_between_body_weight_and_linear_bo dy_measurements_in_the_Cobb_broiler_chicken.

[29] S. L. Weimer, A. Mauromoustakos, D. M. Karcher, and M. A. Erasmus, "Differences in performance,body conformation, and welfare of conventional and slow-growing broiler chickens raised at 2 stocking densities," *Poultry Science*, vol. 99, no. 9, pp. 4398–4407, Sep. 2020, doi: 10.1016/j.psj.2020.06.009. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7598105/.

[30] A. K. Das, S. Kumar, and A. Rahim, "Genetics of Body Conformation and Feed EfficiencyCharacteristics in a Control Line of Rhode Island Red Chicken," *Iranian Journal of Applied Animal Science*, vol. 5, no. 4, pp. 965–973, Dec. 2015 [Online]. Available: https://journals.iau.ir/article_516636.html.

[31] R. Grillo, "Chicken Behavior: An Overview of Recent Science," *Free From Harm*, Oct. 29, 2015. [Online]. Available: https://freefromharm.org/chicken-behavior-an-overview-of-recent-science/.

[32] L. Costa, D. Pereira, L. Bueno, and H. Pandorfi, “Some aspects of chicken behavior and welfare,”*Revista Brasileira de Ciência Avícola*, vol. 14, no. 3, pp. 159–164, Sep. 2012, doi: 10.1590/s1516-635x2012000300001. [Online]. Available: https://www.scielo.br/j/rbca/a/L3JqQJCP3QrBGBySsgscvFN/?lang=en.

[33] Dr. Jacquie Jacob, “NORMAL BEHAVIORS OF CHICKENS IN SMALL AND BACKYARD POULTRY FLOCKS,” 2023, [Online]. Available: https://poultry.extension.org/articles/poultry-behavior/normal-behaviors-of-chickens-in-small-and-backyard-poultry-flocks/

[34]] LearnPoultry, “Why do Chickens Flap Their Wings?,” 2023, [Online]. Available: https://learnpoultry.com/chickens-flap-wings/

[35] “Chickens & Heat Stress: How to Identify, Prevent, & Treat,” *Grubblyfarms.com*, 2021. [Online]. Available: https://grubblyfarms.com/blogs/the-flyer/chickens-and-heat-stress#:~:text=Here%27s%20some%20tell%2 Dtale%20signs,beginning%20signs%20of%20heat%20stress.

[36] Dr Colin Tidy, “Hyperventilation (Causes, Symptoms and Treatment),” *Patient.info*, Nov. 24, 2021. [Online]. Available: https://patient.info/doctor/hyperventilation.

[37] J. Cvt, “Poultry Medicine 101” [Online]. Available: https://www.isvma.org/wp-content/uploads/2018/10/Poultry-ISVMA-.pdf. [Accessed: Jun. 26, 2022]

[38] L. C. Dawson, T. M. Widowski, Z. Liu, A. M. Edwards, and S. Torrey, “In pursuit of a better broiler: comparison of the inactivity, behavior, and enrichment use of fast- and slower growing broiler chickens,” *Poultry Science*, vol. 100, no. 12, p. 101451,

Dec. 2021, doi: 10.1016/j.psj.2021.101451. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0032579121004740.

[39] Dr. Jose A. Linares, "Lice Infestation in Chickens," 2023, [Online]. Available: https://www.hobbyfarms.com/lice-infestation-in-chickens-2/

[40] A. Reddy, P. Praveen, P. Prasa-Dini, and A. Anuradha, "ASSESSMENT OF AIR QUALITY IN POULTRY HOUSE," *Industrial Pollution Control*, vol. 23, no. 2, pp. 369–372, 2007 [Online]. Available: https://www.icontrolpollution.com/articles/assessment-of-air-quality-in-a-poultryhouse-.pdf.

[41] Amit, "Managemental strategies to reduce air pollution from poultry," *ENGLISH MONTHLY MAGAZINE*, May 18, 2019. [Online]. Available: https://thepoultrypunch.com/2019/05/managemental-strategies-to-reduce-air-pollution-from-poultry/.

[42] J. Lloret, "IEEE Sensors Council CALL FOR PAPERS IEEE Sensors Journal Special issue on RGB-DSensors and 3D Reconstruction" [Online]. Available: https://ieee-sensors.org/wp-content/uploads/2019/05/SJ_Special_Issue_RGB-D_Sensors.pdf

[43] K. Yoshimoto and M. Shinya, "Use of the Azure Kinect to measure foot clearance during obstacle crossing: A validation study," *PLOS ONE*, vol. 17, no. 3, p. e0265215, Mar. 2022, doi: 10.1371/journal.pone.0265215. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0265215.

[44] Andrea De Montis, A. Pinna, M. Barra, and E. Vranken, “Analysis of poultry eating and drinking behavior by software eYeNamic,” *ResearchGate*, Jun. 09, 2013. [Online]. Available: https://www.researchgate.net/publication/269651570_Analysis_of_poultry_eating_and_drinking_behavior _by_software_eYeNamic.

[45] A. Sani and E. A. Marlin, “Body Mass Index Measurement System using Image Processing,” no. Icae2020, pp. 81–84, 2021, doi: 10.5220/0010352000810084.https://www.scitepress.org/Papers/2020/103520/103520.pdf

[46] Jhuria, M., Kumar, A., & Borse, R. (2013). Image processing for smart farming: Detection of disease and fruit grading. 2013 IEEE 2nd International Conference on Image Information Processing, IEEE ICIIP 2013, 521–526. https://doi.org/10.1109/ICIIP.2013.6707647

[47] Kashiha, M. A., Bahr, C., Vranken, E., Hong, S., & Berckmans, D. (2014). Monitoring System to Detect Problems in Broiler Houses Based on Image Processing. Proceedings of the International Conference of Agricultural Engineering, 2003, 6–10. www.eurageng.eu

[48] Shuai, S., Ling, Y., Shihao, L., Haojie, Z., Xuhong, T., Caixing, L., Aidong, S., & Hanxing, L. (2020). Research on 3D surface reconstruction and body size measurement of pigs based on multi-view RGB-D cameras. Computers and Electronics in Agriculture, 175(January), 105543. https://doi.org/10.1016/j.compag.2020.105543

[49] CÖMERT, O., HEKİM, M., & ADEM, K. (2017). Weight and Diameter Estimation Using Image Processing and Machine Learning Techniques on Apple Images. Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi, 147–154. https://doi.org/10.29137/umagd.350588

[50] M. Brain and C. Pollette, "How Microprocessors Work," *HowStuffWorks*, Apr. 2000. [Online]. Available: https://computer.howstuffworks.com/microprocessor.htm.

[51] "US4492450A - Camera with microprocessor - Google Patents," *Google.com*, Sep. 27, 1982. [Online]. Available: https://patents.google.com/patent/US4492450.

[52] "What are the differences between Raspberry Pi and Arduino?," *Electronics Hub*, Apr. 05, 2021. [Online]. Available: https://www.electronicshub.org/raspberry-pi-vs-arduino/#Raspberry_Pi.

[53] "Data Communication & Computer Network," *Tutorialspoint.com*, 2022. [Online]. Available: https://www.tutorialspoint.com/data_communication_computer_network/index.htm.

[54] TechTarget Contributor, "Web application (Web app)," *SearchSoftwareQuality*, 2019. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app#:~:text=A%20Web %20application%20(Web%20app,all%2C%20websites%20contain%20Web%20apps ..

[55] "What is the Internet of Things (IoT)?," *Oracle.com*, 2020. [Online]. Available: https://www.oracle.com/ph/internet-of-things/what-is-iot/.

[56] H.-D. Ma, “Internet of Things: Objectives and Scientific Challenges,” *Journal of Computer Science and Technology*, vol. 26, no. 6, pp. 919–924, Nov. 2011, doi: 10.1007/s11390-011-1189-5. [Online]. Available: https://link.springer.com/article/10.1007/s11390-011-1189-5.

[57] “Websites Hosting,” *Tutorialspoint.com*, 2022. [Online]. Available: https://www.tutorialspoint.com/internet_technologies/website_hosting.htm.

[58] “What is Shared Hosting? | Domain.com | Blog,” *Domain.com | Blog*, Mar. 12, 2019. [Online]. Available: https://www.domain.com/blog/what-is-shared-hosting/.

[59] “What is Shared Hosting? | HostGator Support,” *Hostgator.com*, 2022. [Online]. Available: https://www.hostgator.com/help/article/what-is-shared-hosting.

[60] IBM Cloud Education, “vps,” *Ibm.com*, Sep. 16, 2019. [Online]. Available: https://www.ibm.com/cloud/learn/vps#:~:text=A%20VPS%2C%20or%20virtual%20private,provider%2C% 20that%20runs%20multiple%20VPSs..

[61] N. Sharma, “An Explanation - Why VPS Hosting Is Important,” *YourStory.com*, Nov. 28, 2018. [Online]. Available: https://yourstory.com/mystory/an-explanation-why-vps-hosting-is-important-4qyvyiis9z/amp.

[62] “What is Cloud Hosting,” *Atlantic.net*, Feb. 24, 2022. [Online]. Available: https://www.atlantic.net/vps-hosting/what-is-cloud-hosting/.

[63] “What is cloud hosting,” *Ibm.com*, 2021. [Online]. Available: https://www.ibm.com/cloud/learn/what-is-cloud-hosting#:~:text=Cloud%20hosting%20makes%20applications%20and,ensuring%20greater%20flexibility%20and%20scalability..

[64] "Rackspace," *Rackspace Technology*, 2021. [Online]. Available: https://www.rackspace.com/en-ph/library/what-is-dedicated-hosting#:~:text=Dedicated%20hosting%20is %20an%20Internet,requirements%2C%20including%20performance%20and%20security..

[65] https://www.facebook.com/inmotionhosting, "Who Needs Dedicated Hosting – And Why?," *Search Engine Journal*, Dec. 21, 2021. [Online]. Available: https://www.searchenginejournal.com/inmotion-need-dedicated-hosting/429080/.

[66] Techopedia, "Web Programming," *Techopedia.com*, Feb. 27, 2012. [Online]. Available: https://www.techopedia.com/definition/23898/web-programming#:~:text=Web%20programming%20refers %20to%20the,JavaScript%2C%20Perl%205%20and%20PHP..

[67] "Popular programming languages for web development," *Analytics India Magazine*, May 16, 2022. https://analyticsindiamag.com/popular-programming-languages-for-web-development/

[68] "Comparing Rust and C++ in IOT Development," *Engineering Education (EngEd) Program | Section*, 2021. [Online]. Available: https://www.section.io/engineering-education/iot-development-with-rust-vs-c++/#:~:text=IoT%20development%20with%20C%2B%2B&text=C%2B%2B%20is%20a%20middle%2Dlevel,speed%2C%20efficiency% 2C%20and%20performance..

[69] M. Kelly, “The Case for C++ in the IoT World,” *Embarcadero RAD Studio, Delphi, & C++Builder Blogs*, Jun. 14, 2018. [Online]. Available: https://blogs.embarcadero.com/the-case-for-c-in-the-iot-world/.

[70] L. K. S. Tolentino *et al.*, “Weight prediction system for Nile Tilapia using image processing and predictive analysis,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 399–406, 2020, doi: 10.14569/IJACSA.2020.0110851.

[71] C. Bhatt, A. Hassanien, N. A. Shah, and J. Thik, “Barqi Breed Sheep Weight Estimation based on Neural Network with Regression,” 2018. [Online]. Available: http://arxiv.org/abs/1807.10568

[72] M. F. Hansen, M. L. Smith, L. N. Smith, K. Abdul Jabbar, and D. Forbes, “Automated monitoring of dairy cow body condition, mobility and weight using a single 3D video capture device,” Comput. Ind., vol. 98, pp. 14–22, 2018, doi: 10.1016/j.compind.2018.02.011.

[73] M. Gjergji *et al.*, “Deep Learning Techniques for Beef Cattle Body Weight Prediction,” *Proc. Int. Jt. Conf. Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9207624.

[74] A. Lina Zhang, B. Pei Wu, C. Tana Wuyun, D. Xinhua Jiang, E. Chuanzhong Xuan, and F. YanhuaMa, “Algorithm of sheep body dimension measurement and its applications based on image analysis,” *Comput. Electron. Agric.*, vol. 153, no. July, pp. 33–45, 2018, doi: 10.1016/j.compag.2018.07.033.

[75] Y. Cang, H. He, and Y. Qiao, “An Intelligent Pig Weights Estimate Method Based on Deep Learning inSow Stall Environments,” *IEEE Access*, vol. 7, pp. 164867–164875, 2019, doi: 10.1109/ACCESS.2019.2953099.

[76] P. Lieng and K. Sangpradit, “Study on duck weight estimation by using image processing,” *E3S Web Conf.*, vol. 187, pp. 0–5, 2020, doi: 10.1051/e3sconf/202018702001.

ANNEXES

ANNEX A

Program Codes

Source Code

flaskapp.py

```
try:
   from pithermalcam import pithermalcam
except:
   from pi_therm_cam import pithermalcam
from flask import Response, request
import threading
import time, socket, logging, traceback
import matplotlib.pyplot as plt
from flask import Flask, render_template, Response, request, session, redirect, send_file,
jsonify
from flask_wtf import FlaskForm
from wtforms import FileField, SubmitField
from wtforms.validators import InputRequired
from werkzeug.utils import secure_filename
import os
import cv2
import csv
from YOLO_Video import video_detection
from YOLO_Video1 import video_detection1
from YOLO_Video2 import video_detection2
from YOLO_Video3 import video_detection3
from YOLO_Video5 import  WebcamVideoStream
import RPi.GPIO as GPIO
import time
import datetime
import numpy as np
import pandas as pd
from flask_login import UserMixin
from sqlalchemy.sql import func
import adafruit_mlx90640
import board
from werkzeug.security import generate_password_hash, check_password_hash
from flask_login import login_user, login_required, logout_user, current_user,
LoginManager
from flask import Blueprint, flash, redirect, url_for
from models import User
from flask_sqlalchemy import SQLAlchemy
from os import path
from collections import Counter
logging.basicConfig(filename='pithermcam.log', filemode='a',
              format='%(asctime)s %(levelname)-8s [%(filename)s:%(name)s:%(lineno)d]
%(message)s',
              level=logging.WARNING, datefmt='%d-%b-%y %H:%M:%S')
```

```
logger = logging.getLogger(__name__)
outputFrame = None
thermcam = None
lock = threading.Lock()
app = Flask(__name__)
app.config['SECRET_KEY'] = 'galluspy'
app.config['UPLOAD_FOLDER'] = 'static/files'
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///{}'.format(os.path.join(os.path.dirname(__file__), 'database.db'))
db = SQLAlchemy(app)
DB_NAME = "database.db"
login_manager = LoginManager()
login_manager.login_view = 'login'
login_manager.init_app(app)
class Note(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    data = db.Column(db.String(10000))
    date = db.Column(db.DateTime(timezone=True), default=func.now())
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(150), unique=True)
    password = db.Column(db.String(150))
    first_name = db.Column(db.String(150))
    notes = db.relationship('Note', backref='user')
    def get_id(self):
        return str(self.id)
@login_manager.user_loader
def load_user(id):
    return User.query.get(int(id))
def create_database():
    with app.app_context():
        if path.exists(DB_NAME):
            db.drop_all()
        db.create_all()
        print('Created Database!')
create_database()

#Login
@app.route('/', methods=['GET', 'POST'])
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form.get('email')
        password = request.form.get('password')
        user = User.query.filter_by(email=email).first()
```

```
        if user:
            if check_password_hash(user.password, password):
                flash('Logged in successfully!', category='success')
                login_user(user, remember=True)
                return redirect(url_for('home'))
            else:
                flash('Incorrect password, try again.', category='error')
        else:
            flash('Email does not exist.', category='error')
    return render_template("login.html", user=current_user)
@app.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('login'))

#Sign-Up
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        email = request.form.get('email')
        first_name = request.form.get('fullName')
        password1 = request.form.get('password1')
        password2 = request.form.get('password2')
        user = User.query.filter_by(email=email).first()
        if user:
            flash('Email already exists.', category='error')
        elif len(email) < 4:
            flash('Email must be greater than 3 characters.', category='error')
        elif len(first_name) < 2:
            flash('First name must be greater than 1 character.', category='error')
        elif password1 != password2:
            flash('Passwords don\'t match.', category='error')
        elif len(password1) < 6:
            flash('Password must be at least 6 characters.', category='error')
        else:
            new_user = User(email=email, first_name=first_name,
password=generate_password_hash(
                password1, method='sha256'))
            db.session.add(new_user)
            db.session.commit()
            login_user(new_user, remember=True)
            flash('Account created!', category='success')
            return redirect(url_for('home'))
    return render_template("signup.html", user=current_user)
class UploadFileForm(FlaskForm):
```

```
    file = FileField("File",validators=[InputRequired()])
    submit = SubmitField("Run")
def generate_frames():
    yolo_output = video_detection()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames_web():
    yolo_output = video_detection()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames1():
    yolo_output = video_detection1()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames_web1():
    yolo_output = video_detection1()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```

```
def generate_frames2():
    yolo_output = video_detection2()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames_web2():
    yolo_output = video_detection2()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames3():
    yolo_output = video_detection3()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
def generate_frames_web3():
    yolo_output = video_detection3()
    for detection_ in yolo_output:
        if detection_ is None:
            continue
        ref, buffer = cv2.imencode('.jpg', detection_)
        if not ref:
            continue
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```

#System Homepage

```
@login_required
@app.route('/home', methods=['GET','POST'])
def home():
    session.clear()
    return render_template('indexproject.html')
@app.route("/webcam", methods=['GET','POST'])
def webcam():
    session.clear()
    return render_template('ui.html')
@app.route("/webcam2", methods=['GET','POST'])
def webcam2():
    session.clear()
    return render_template('ui2.html')
@app.route("/webcam3", methods=['GET','POST'])
def webcam3():
    session.clear()
    return render_template('ui3.html')
@app.route("/webcam4", methods=['GET','POST'])
def webcam4():
    session.clear()
    return render_template('ui4.html')
@app.route('/FrontPage', methods=['GET','POST'])
def front():
    form = UploadFileForm()
    if form.validate_on_submit():
        file = form.file.data
        file.save(os.path.join(os.path.abspath(os.path.dirname(__file__)),
app.config['UPLOAD_FOLDER'],
                               secure_filename(file.filename)))
        session['video_path'] = os.path.join(os.path.abspath(os.path.dirname(__file__)),
app.config['UPLOAD_FOLDER'],
                                             secure_filename(file.filename))
    return render_template('videoprojectnew.html', form=form)
@app.route('/video')
def video():
    return Response(generate_frames(path_x = session.get('video_path',
None)),mimetype='multipart/x-mixed-replace; boundary=frame')
@app.route('/webapp')
def webapp():
    return Response(generate_frames_web(),mimetype='multipart/x-mixed-replace;
boundary=frame')
@app.route('/webapp2')
def webapp2():
    return Response(generate_frames_web1(),mimetype='multipart/x-mixed-replace;
boundary=frame')
@app.route('/webapp3')
```

```
def webapp3():
   return Response(generate_frames_web2(),mimetype='multipart/x-mixed-replace;
boundary=frame')
@app.route('/webapp4')
def webapp4():
   return Response(generate_frames_web3(),mimetype='multipart/x-mixed-replace;
boundary=frame')
records = []
@app.route("/")
def index():
   return render_template("ui2.html", records=records)
@app.route("/add_record", methods=["POST"])
def add_record():
   time = request.form["time"]
   active = int(request.form["active"])
   deceased = int(request.form["deceased"])
   records.append({"time": time, "active": active, "deceased": deceased})
   return render_template("ui2.html", records=records)
@app.route('/prevt')
def prevt():
   session.clear()
   return render_template("prevt.html")
@app.route('/preva')
def preva():
   session.clear()
   return render_template("preva.html")
@app.route('/prevv')
def prevv():
   session.clear()
   return render_template("prevv.html")
@app.route('/data')
def data():
   session.clear()
   return render_template("data.html")
@app.route('/indextemp1')
def indextemp1():
   session.clear()
   return render_template("indextemp.html")
@app.route('/save')
def save_image():
      thermcam.save_image()
      return ("Snapshot Saved")
@app.route('/units')
def change_units():
      thermcam.use_f = not thermcam.use_f
      return ("Units changed")
```

```
@app.route('/colormap')
def increment_colormap():
        thermcam.change_colormap()
        return ("Colormap changed")
@app.route('/colormapback')
def decrement_colormap():
        thermcam.change_colormap(forward=False)
        return ("Colormap changed back")
@app.route('/filter')
def toggle_filter():
        thermcam.filter_image=not thermcam.filter_image
        return ("Filtering Toggled")
@app.route('/interpolation')
def increment_interpolation():
        thermcam.change_interpolation()
        return ("Interpolation Changed")
@app.route('/interpolationback')
def decrement_interpolation():
        thermcam.change_interpolation(forward=False)
        return ("Interpolation Changed Back")
@app.route('/exit')
def appexit():
        global thermcam
        func = request.environ.get('werkzeug.server.shutdown')
        if func is None:
                raise RuntimeError('Not running with the Werkzeug Server')
        func()
        thermcam = None
        return 'Server shutting down...'
@app.route("/video_feed")
def video_feed():
        return Response(generate(), mimetype="multipart/x-mixed-replace;
boundary=frame")
def gen(camera):
    while True:
        if camera.stopped:
            break
        frame = camera.read()
        ret, jpeg = cv2.imencode('.jpg',frame)
        if jpeg is not None:
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n\r\n')
        else:
            print("frame is none")
@app.route('/video_feed2')
def video_feed2():
```

```
    return Response(gen(WebcamVideoStream().start()),
                mimetype='multipart/x-mixed-replace; boundary=frame')
def get_ip_address():
        """Find the current IP address of the device"""
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        s.connect(("8.8.8.8", 80))
        ip_address=s.getsockname()[0]
        s.close()
        return ip_address
def pull_images():
        global thermcam, outputFrame
        while thermcam is not None:
                current_frame=None
                try:
                        current_frame = thermcam.update_image_frame()
                except Exception:
                        print("Too many retries error caught; continuing...")
                        logger.info(traceback.format_exc())
                if current_frame is not None:
                        with lock:
                                outputFrame = current_frame.copy()
def generate():
        global outputFrame, lock
        while True:
                with lock:
                        if outputFrame is None:
                                continue
                        (flag, encodedImage) = cv2.imencode(".jpg", outputFrame)
                        if not flag:
                                continue
                yield(b'--frame\r\n' b'Content-Type: image/jpeg\r\n\r\n' +
bytearray(encodedImage) + b'\r\n')
def start_server(output_folder:str = '/home/pi/pithermalcam/saved_snapshots/'):
        global thermcam
        thermcam = pithermalcam(output_folder=output_folder)
        time.sleep(0.1)
        t = threading.Thread(target=pull_images)
        t.daemon = True
        t.start()
        ip=get_ip_address()
        port=8000
        print(f'Server can be found at {ip}:{port}')
        app.run(host=ip, port=port, debug=True,threaded=True, use_reloader=False)
```

#Ammonia Monitoring Code

```
GPIO.setmode(GPIO.BCM)
```

```
MQ_PIN = 4
history_length = 10
ammonia_history = []
RO_CLEAN_AIR_FACTOR = 9.83
CALIBRATION_SAMPLE_TIMES = 50
CALIBRATION_SAMPLE_INTERVAL = 0.1
MIN_AMMONIA_LEVEL = 10
MAX_AMMONIA_LEVEL = 50
RESISTIVITY_ADJUSTMENT_FACTOR = 2.0
CALIBRATION_AMMONIA_LEVEL = 20
csv_file_path = os.path.join(os.path.dirname(os.path.abspath(__file__)),
"ammonia_history.csv")
@app.route("/mqchart")
def mqchart():
  return render_template('mqcharts.html')
@app.route("/get_ammonia")
def get_ammonia():
  global ammonia_history
  ammonia_level = read_ammonia_level()
  current_time = time.strftime("%H:%M:%S")
  current_date = datetime.datetime.now().strftime("%Y-%m-%d")
  ammonia_reading = {"time": current_time, "ammonia": ammonia_level, "date":
current_date}
  ammonia_history.append(ammonia_reading)
  ammonia_history = ammonia_history[-history_length:]
  plot_graph()
  save_to_csv()
  return jsonify(ammonia_history)
def read_ammonia_level():
  GPIO.setup(MQ_PIN, GPIO.OUT)
  GPIO.output(MQ_PIN, GPIO.LOW)
  time.sleep(0.1)
  GPIO.setup(MQ_PIN, GPIO.IN)
  time.sleep(0.2)
  rs_air = 0
  for _ in range(CALIBRATION_SAMPLE_TIMES):
    rs_air += MQResistanceCalculation(GPIO.input(MQ_PIN))
    time.sleep(CALIBRATION_SAMPLE_INTERVAL)
  rs_air /= CALIBRATION_SAMPLE_TIMES
  rs = 0
  for _ in range(CALIBRATION_SAMPLE_TIMES):
    rs += MQResistanceCalculation(GPIO.input(MQ_PIN))
    time.sleep(CALIBRATION_SAMPLE_INTERVAL)
  rs /= CALIBRATION_SAMPLE_TIMES
  ratio = rs / rs_air
  ammonia_level = 10 ** ((np.log10(ratio) - 1.0179) / -0.3357)
```

```
    surrounding_ammonia_level = get_surrounding_ammonia_level()
    ammonia_level += surrounding_ammonia_level
    calibration_factor = CALIBRATION_AMMONIA_LEVEL / (ammonia_level - 
surrounding_ammonia_level)
    rs *= calibration_factor
    adjusted_resistance = rs * RESISTIVITY_ADJUSTMENT_FACTOR
    ammonia_level = calculate_ammonia_level(adjusted_resistance)
    ammonia_level = max(MIN_AMMONIA_LEVEL, min(MAX_AMMONIA_LEVEL, 
ammonia_level))
    return ammonia_level
def MQResistanceCalculation(raw_adc):
    if raw_adc == 0:
        return 0
    resistance = (float(1023) / raw_adc) - 1
    return resistance
def calculate_ammonia_level(resistance):
    ratio = resistance / RO_CLEAN_AIR_FACTOR
    ammonia_level = 10 ** ((np.log10(ratio) - 1.0179) / -0.3357)
    return ammonia_level
def get_surrounding_ammonia_level():
    surrounding_levels = [4.5, 6.2, 5.8, 7.1, 6.5]
    average_level = np.mean(surrounding_levels)
    return average_level
def plot_graph():
    times = [reading["time"] for reading in ammonia_history]
    levels = [reading["ammonia"] for reading in ammonia_history]
    plt.plot(times, levels)
    plt.xlabel("Time")
    plt.ylabel("Ammonia Level (PPM)")
    plt.title("Ammonia Level Over Time")
    plt.xticks(rotation=45)
    plt.savefig(os.path.join(os.path.dirname(os.path.abspath(__file__)), "static", 
"ammonia_graph.png"))
    plt.clf()
def save_to_csv():
    with open(csv_file_path, 'a', newline='') as csvfile:
        fieldnames = ['Time', 'Ammonia Level', 'Date']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        if csvfile.tell() == 0:
            writer.writeheader()
        for reading in ammonia_history:
            time = reading['time']
            level = '{:.3f}'.format(reading['ammonia'])
            date = datetime.datetime.now().strftime("%Y-%m-%d")
            writer.writerow({'Time': time, 'Ammonia Level': level, 'Date': date})
@app.route("/download_ammn_csv")
```

```
def download_ammn_csv():
    return send_file(csv_file_path, as_attachment=True,
download_name="ammonia_history.csv")
```

#Temperature Monitoring Code

```
i2c = board.I2C()
mlx = adafruit_mlx90640.MLX90640(i2c)
mlx_shape = (24, 32)
frame = [0] * (mlx_shape[0] * mlx_shape[1])
csv_filename = os.path.join(os.path.dirname(__file__), "temperature_history.csv")
temperature_history = []
def load_temperature_history_from_csv(filename):
    data = []
    with open(filename, "r", newline="") as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            data.append({
                "time": row["time"],
                "temperature": [float(temp) for temp in row["average_temperature"].split(",")]
            })
    return data
load_temperature_history_from_csv(csv_filename)
@app.route("/chart")
def chart():
    return render_template('charts.html')
@app.route("/get_temperature")
def get_temperature():
    global temperature_history
    try:
        mlx.getFrame(frame)
    except ValueError:
        pass
    frame_data = np.fliplr(np.reshape(frame, mlx_shape)).T
    temperatures = frame_data.flatten()
    current_time = time.strftime("%H:%M:%S")
    current_date = datetime.datetime.now().strftime("%Y-%m-%d")
    temperature_reading = {"time": current_time, "temperature": temperatures.tolist(),
"date": current_date}
    temperature_history.append(temperature_reading)
    temperature_history = temperature_history[-10:]
    save_temperature_history_to_csv(temperature_history, csv_filename)
    return jsonify(temperature_history)
def save_temperature_history_to_csv(temperature_history, filename):
    with open(filename, "a", newline="") as csvfile:
        fieldnames = ["date", "time", "average_temperature"]
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
```

```
        for reading in temperature_history:
            temperatures = reading["temperature"]
            average_temperature = round(np.mean(temperatures), 3)
            current_date = datetime.date.today().strftime("%Y-%m-%d")
            writer.writerow({"date": current_date, "time": reading["time"],
"average_temperature": average_temperature})
@app.route("/download_temp_csv")
def download_temp_csv():
    return send_file(csv_filename, as_attachment=True)
if __name__ == '__main__':
            start_server()
```

YOLO_Video.py

#Mortality Rate Monitoring Code

```
from ultralytics import YOLO
import cv2
import math
def video_detection():
    cap = cv2.VideoCapture('/home/galluspy/Downloads/chicken3.mp4')
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    new_width = int(frame_width / 2)
    new_height = int(frame_height / 2)
    model = YOLO("/home/galluspy/Downloads/best2.pt")
    classNames = ["active","deceased"]
    while True:
        success, img = cap.read()
        if not success:
            break
        img = cv2.resize(img, (new_width, new_height))
        results = model(img, stream=True)
        for r in results:
            boxes = r.boxes
            for box in boxes:
                x1, y1, x2, y2 = box.xyxy[0]
                x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
                print(x1, y1, x2, y2)
                cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)
                conf = math.ceil((box.conf[0] * 100)) / 100
                cls = int(box.cls[0])
                class_name = classNames[cls]
                label = class_name
```

```
            t_size = cv2.getTextSize(label, 0, fontScale=1, thickness=2)[0]
            print(t_size)
            c2 = x1 + t_size[0], y1 - t_size[1] - 3
            cv2.rectangle(img, (x1, y1), c2, [255, 0, 255], -1, cv2.LINE_AA)
            cv2.putText(img, label, (x1, y1 - 2), 0, 1, [255, 255, 255], thickness=1,
lineType=cv2.LINE_AA)
        yield img
    cv2.destroyAllWindows()
```

YOLO_Video1.py

#Mortality Rate Graph Code

```
import matplotlib
matplotlib.use('Agg')
from datetime import datetime
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt
import numpy as np
def video_detection1():
    cap = cv2.VideoCapture('/home/galluspy/Downloads/chicken3.mp4')
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    model = YOLO("/home/galluspy/Downloads/best2.pt")
    classNames = ["active", "deceased"]
    classCounts = {class_name: 0 for class_name in classNames}
    fig, ax = plt.subplots()
    x = []
    y_active = []
    y_deceased = []
    line_active, = ax.plot(x, y_active, label='Active')
    line_deceased, = ax.plot(x, y_deceased, label='Deceased')
    ax.legend()
    line_chart_height = frame_height - 100
    while True:
        success, img = cap.read()
        results = model(img, stream=True)
        cv2.rectangle(img, (0, 0), (frame_width, 100), (255, 255, 255), -1)
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        cv2.putText(img, current_time, (20, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
0, 0), 2)
        classCounts = {class_name: 0 for class_name in classNames}
        for r in results:
            boxes = r.boxes
            for box in boxes:
                cls = int(box.cls[0])
                class_name = classNames[cls]
```

```
            classCounts[class_name] += 1
        x.append(datetime.now())
        y_active.append(classCounts["active"])
        y_deceased.append(classCounts["deceased"])
        line_active.set_data(x, y_active)
        line_deceased.set_data(x, y_deceased)
        ax.set_xlim([min(x), max(x)])
        ax.set_ylim([0, max(max(y_active), max(y_deceased)) + 10])
        plt.draw()
        plt.savefig('/home/galluspy/Downloads/line_chart.png')
        line_chart_img = cv2.imread('/home/galluspy/Downloads/line_chart.png')
        line_chart_img = cv2.resize(line_chart_img, (frame_width, line_chart_height))
        img[100:frame_height, 0:frame_width] = line_chart_img
        plt.draw()
        plt.pause(0.01)
        yield img
    cv2.destroyAllWindows()
```

YOLO_Video2.py

#Live Weight Monitoring Code

```
from ultralytics import YOLO
import cv2
import math
def video_detection2():
    cap = cv2.VideoCapture('/home/galluspy/Downloads/chicken1.mp4')
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    new_width = int(frame_width / 2)
    new_height = int(frame_height / 2)
    model = YOLO("/home/galluspy/Downloads/best2.pt")
    classNames = ["0.5", "0.6", "0.7", "0.8", "0.9", "1.0", "1.1", "1.2", "1.3", "1.4", "1.5",
"1.6", "1.7", "1.8", "1.9", "2"]
    while True:
        success, img = cap.read()
        if not success:
            break
        img = cv2.resize(img, (new_width, new_height))
        results = model(img, stream=True)
        for r in results:
            boxes = r.boxes
            for box in boxes:
                x1, y1, x2, y2 = box.xyxy[0]
                x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
                print(x1, y1, x2, y2)
                cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)
                conf = math.ceil((box.conf[0] * 100)) / 100
```

```
            cls = int(box.cls[0])
            class_name = classNames[cls]
            label = class_name
            t_size = cv2.getTextSize(label, 0, fontScale=1, thickness=2)[0]
            print(t_size)
            c2 = x1 + t_size[0], y1 - t_size[1] - 3
            cv2.rectangle(img, (x1, y1), c2, [255, 0, 255], -1, cv2.LINE_AA)
            cv2.putText(img, label, (x1, y1 - 2), 0, 1, [255, 255, 255], thickness=1,
lineType=cv2.LINE_AA)
        yield img
    cv2.destroyAllWindows()
```

YOLO_Video3.py
#Live Weight Graph Code

```
import matplotlib
matplotlib.use('Agg')
from datetime import datetime
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt
import numpy as np
def video_detection3():
    cap = cv2.VideoCapture('/home/galluspy/Downloads/chicken1.mp4')
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    model = YOLO("/home/galluspy/Downloads/best2.pt")
    classNames = [0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2]
    classCounts = {class_name: 0 for class_name in classNames}
    fig, ax = plt.subplots()
    x = []
    y_A = []
    y_B = []
    y_C = []
    y_D = []
    y_E = []
    y_F = []
    y_G = []
    y_H = []
    y_I = []
    y_J = []
    y_K = []
    y_L = []
    line_A, = ax.plot(x, y_A, label='0.5')
    line_B, = ax.plot(x, y_B, label='0.6')
    line_C, = ax.plot(x, y_C, label='0.7')
    line_D, = ax.plot(x, y_D, label='0.8')
```

```
    line_E, = ax.plot(x, y_E, label='0.9')
    line_F, = ax.plot(x, y_F, label='1.0')
    line_G, = ax.plot(x, y_G, label='1.1')
    line_H, = ax.plot(x, y_H, label='1.2')
    line_I, = ax.plot(x, y_I, label='1.3')
    line_J, = ax.plot(x, y_J, label='1.4')
    line_K, = ax.plot(x, y_K, label='1.5')
    line_L, = ax.plot(x, y_L, label='1.6')
    ax.legend()
    line_chart_height = frame_height - 100
    while True:
        success, img = cap.read()
        results = model(img, stream=True)
        cv2.rectangle(img, (0, 0), (frame_width, 100), (255, 255, 255), -1)
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        cv2.putText(img, current_time, (20, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
0, 0), 2)
        for r in results:
            boxes = r.boxes
            for box in boxes:
                cls = int(box.cls[0])
                class_name = classNames[cls]
                classCounts[class_name] += 1
        x.append(datetime.now())
        y_A.append(classCounts[0.5])
        y_B.append(classCounts[0.6])
        y_C.append(classCounts[0.7])
        y_D.append(classCounts[0.8])
        y_E.append(classCounts[0.9])
        y_F.append(classCounts[1.0])
        y_G.append(classCounts[1.1])
        y_H.append(classCounts[1.2])
        y_I.append(classCounts[1.3])
        y_J.append(classCounts[1.4])
        y_K.append(classCounts[1.5])
        y_L.append(classCounts[1.6])
        line_A.set_data(x, y_A)
        line_B.set_data(x, y_B)
        line_C.set_data(x, y_C)
        line_D.set_data(x, y_D)
        line_E.set_data(x, y_E)
        line_F.set_data(x, y_F)
        line_G.set_data(x, y_G)
        line_H.set_data(x, y_H)
        line_I.set_data(x, y_I)
        line_J.set_data(x, y_J)
```

```
        line_K.set_data(x, y_K)
        line_L.set_data(x, y_L)
        ax.set_xlim([min(x), max(x)])
        ax.set_ylim([0, max(max(y_A), max(y_B), max(y_C), max(y_D), max(y_E),
max(y_F), max(y_G), max(y_H), max(y_I), max(y_J), max(y_K), max(y_L)) + 10])
        plt.draw()
        plt.savefig('/home/galluspy/Downloads/line_chart.png')
        line_chart_img = cv2.imread('/home/galluspy/Downloads/line_chart.png')
        line_chart_img = cv2.resize(line_chart_img, (frame_width, line_chart_height))
        img[100:frame_height, 0:frame_width] = line_chart_img
        classCounts = {class_name: 0 for class_name in classNames}
        plt.draw()
        plt.pause(0.01)
        yield img
    cv2.destroyAllWindows()
```

YOLO_Video5.py

#Live Webcam Code

```
import cv2
from threading import Thread
import time
import numpy as np
class WebcamVideoStream:
    def __init__(self, src= 0):
        self.stream = cv2.VideoCapture(src)
        (self.grabbed, self.frame) = self.stream.read()
        self.stopped = False
        time.sleep(2.0)
    def start(self):
        t = Thread(target=self.update, args=())
        t.daemon = True
        t.start()
        return self
    def update(self):
        while True:
            if self.stopped:
                return
            (self.grabbed, self.frame) = self.stream.read()
    def read(self):
        return self.frame
    def stop(self):
        self.stopped = True
```

login.html

```
{% extends "base.html" %} {% block title %}Login{% endblock %} {% block content
    %}
```

```
  <form method="POST">
   <h3 align="center" style="color: rgb(241, 241, 241);">Login</h3>
   <div class="form-group">
    <label for="email" style="color: rgb(241, 241, 241);">Email Address</label>
    <input
     type="email"
     class="form-control"
     id="email"
     name="email"
     placeholder="Enter email"
    />
   </div>
   <div class="form-group">
    <label for="password" style="color: rgb(241, 241, 241);">Password</label>
    <input
     type="password"
     class="form-control"
     id="password"
     name="password"
     placeholder="Enter password"
    />
   </div>
   <br />
   <button type="submit" class="btn btn-primary">Login</button>
  </form>
  {% endblock %}
```

signup.html

```
{% extends "base.html" %}
{% block title %}Sign Up{% endblock %}
{% block content %}
<style>
 body {
  background-image: url('../static/images/dlog.jpg');
  background-size: cover;
  background-position: center center;
  background-repeat: no-repeat;
  height: 100vh;
  margin: 0;
  padding: 0;
 }
</style>
<form method="POST">
 <h3 align="center" style="color: rgb(241, 241, 241);">Sign Up</h3>
 <div class="form-group">
  <label for="email" style="color: rgb(241, 241, 241);">Email Address</label>
```

```
    <input
      type="email"
      class="form-control"
      id="email"
      name="email"
      placeholder="Enter email"
    />
  </div>
      <div class="form-group">
        <label for="fullName" style="color: rgb(241, 241, 241);">Full Name</label>
        <input
          type="text"
          class="form-control"
          id="fullName"
          name="fullName"
          placeholder="Enter full name"
        />
      </div>
      <div class="form-group">
        <label for="password1" style="color: rgb(241, 241, 241);">Password</label>
        <input
          type="password"
          class="form-control"
          id="password1"
          name="password1"
          placeholder="Enter password"
        />
      </div>
      <div class="form-group">
        <label for="password2" style="color: rgb(241, 241, 241);">Password
(Confirm)</label>
        <input
          type="password"
          class="form-control"
          id="password2"
          name="password2"
          placeholder="Confirm password"
        />
      </div>
      <br />
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
    {% endblock %}
```

indexproject.html
<!-- Homepage -->

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>GalluSpy Camera</title>
<style>
      @import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@100;300;400;500;700&disp
lay=swap');
:root{
   --main-color:#d3ad7f;
   --black:#13131a;
   --bg:#010103;
   --border:.1rem solid rgba(255,255,255,.3);
}
*{
   font-family: 'Roboto', sans-serif;
   margin:0; padding:0;
   box-sizing: border-box;
   outline: none; border:none;
   text-decoration: none;
   text-transform: capitalize;
   transition: .2s linear;
}
html{
   font-size: 62.5%;
   overflow-x: hidden;
   scroll-padding-top: 9rem;
   scroll-behavior: smooth;
}
html::-webkit-scrollbar{
   width: .8rem;
}
html::-webkit-scrollbar-track{
   background: transparent;
}
html::-webkit-scrollbar-thumb{
   background: #fff;
   border-radius: 5rem;
}
body{
   background: var(--bg);
}
section{
```

```
    padding:2rem 7%;
}
.heading{
    text-align: center;
    color:#fff;
    text-transform: uppercase;
    padding-bottom: 3.5rem;
    font-size: 4rem;
}
.heading span{
    color:var(--main-color);
    text-transform: uppercase;
}
.btn{
    margin-top: 1rem;
    display: inline-block;
    padding:.9rem 3rem;
    font-size: 1.7rem;
    color:#fff;
    background: var(--main-color);
    cursor: pointer;
}
.btn:hover{
    letter-spacing: .2rem;
}
.flip-box {
    background-color: transparent;
    width: 110px;
    height: 100px;
    border: 1px solid #f1f1f1;
    perspective: 1000px;
}
.flip-box-inner {
    position: relative;
    width: 100%;
    height: 100%;
    text-align: center;
    transition: transform 0.8s;
    transform-style: preserve-3d;
}
.flip-box:hover .flip-box-inner {
    transform: rotateY(180deg);
}
.flip-box-front, .flip-box-back {
    position: absolute;
    width: 100%;
```

```
  height: 100%;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}
.flip-box-front {
  background-color: #bbb;
  color: black;
}
.flip-box-back {
  background-color: #555;
  color: white;
  transform: rotateY(180deg);
}
.center {
  line-height: 100px;
  height: 100px;
  border: 3px solid white;
  text-align: center;
}
.center p {
  line-height: 1.5;
  display: inline-block;
  vertical-align: middle;
}
.header{
  background: var(--bg);
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding:1.5rem 7%;
  border-bottom: var(--border);
  position: fixed;
  top:0; left: 0; right: 0;
  z-index: 1000;
}
.header .logo img{
  height: 6rem;
}
.header .navbar a{
  margin:0 1rem;
  font-size: 1.6rem;
  color:#fff;
}
.header .navbar a:hover{
  color:var(--main-color);
  border-bottom: .1rem solid var(--main-color);
```

```
    padding-bottom: .5rem;
}
.header .icons div{
    color:#fff;
    cursor: pointer;
    font-size: 2.5rem;
    margin-left: 2rem;
}
.header .icons div:hover{
    color:var(--main-color);
}
#menu-btn{
    display: none;
}
.header .search-form{
    position: absolute;
    top:115%; right: 7%;
    background: #fff;
    width: 50rem;
    height: 5rem;
    display: flex;
    align-items: center;
    transform: scaleY(0);
    transform-origin: top;
}
.header .search-form.active{
    transform: scaleY(1);
}
.header .search-form input{
    height: 100%;
    width: 100%;
    font-size: 1.6rem;
    color:var(--black);
    padding:1rem;
    text-transform: none;
}
.header .search-form label{
    cursor: pointer;
    font-size: 2.2rem;
    margin-right: 1.5rem;
    color:var(--black);
}
.header .search-form label:hover{
    color:var(--main-color);
}
.header .cart-items-container{
```

```
  position: absolute;
  top:100%; right: -100%;
  height: calc(100vh - 9.5rem);
  width: 35rem;
  background: #fff;
  padding:0 1.5rem;
}
.header .cart-items-container.active{
  right: 0;
}
.header .cart-items-container .cart-item{
  position: relative;
  margin:2rem 0;
  display: flex;
  align-items: center;
  gap:1.5rem;
}
.header .cart-items-container .cart-item .fa-times{
  position: absolute;
  top:1rem; right: 1rem;
  font-size: 2rem;
  cursor: pointer;
  color: var(--black);
}
.header .cart-items-container .cart-item .fa-times:hover{
  color:var(--main-color);
}
.header .cart-items-container .cart-item img{
  height: 7rem;
}
.header .cart-items-container .cart-item .content h3{
  font-size: 2rem;
  color:var(--black);
  padding-bottom: .5rem;
}
.header .cart-items-container .cart-item .content .price{
  font-size: 1.5rem;
  color:var(--main-color);
}
.header .cart-items-container .btn{
  width: 100%;
  text-align: center;
}
.home{
  min-height: 100vh;
  display: flex;
```

```
    align-items: center;
    background:url(../static/images/bg2.jpg) no-repeat;
    background-size: cover;
    background-position: center;
}
.home .content{
    max-width: 60rem;
}
.home .content h3{
    font-size: 6rem;
    text-transform: uppercase;
    color:#fff;
}
.home .content p{
    font-size: 2rem;
    font-weight: lighter;
    line-height: 1.8;
    padding:1rem 0;
    color:#eee;
}
.about .row{
    display: flex;
    align-items: center;
    background:var(--black);
    flex-wrap: wrap;
}
.about .row .image{
    flex:1 1 45rem;
}
.about .row .image img{
    width: 100%;
}
.about .row .content{
    flex:1 1 45rem;
    padding:2rem;
}
.about .row .content h3{
    font-size: 3rem;
    color:#fff;
}
.about .row .content p{
    font-size: 1.6rem;
    color:#ccc;
    padding:1rem 0;
    line-height: 1.8;
}
```

```
.menu .box-container{
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(40rem, 1fr));
    gap:1.5rem;
}
.menu .box-container .box{
    padding:5rem;
    text-align: center;
    border:var(--border);
}
.menu .box-container .box img{
    height: 15rem;
    width: 15rem;
    border-radius: 50%;
    border: 5px solid wheat;
}
.menu .box-container .box h3{
    color: #fff;
    font-size: 2rem;
    padding:1rem 0;
}
.menu .box-container .box .price{
    color: #fff;
    font-size: 2.5rem;
    padding:.5rem 0;
}
.menu .box-container .box .price span{
    font-size: 1.5rem;
    text-decoration: line-through;
    font-weight: lighter;
}
.menu .box-container .box:hover{
    background:#fff;
}
.menu .box-container .box:hover > *{
    color:var(--black);
}
.products .box-container{
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(40rem, 1fr));
    gap:1.5rem;
}
.products .box-container .box{
    text-align: center;
    border:var(--border);
    padding: 2rem;
```

```
}
.products .box-container .box .icons a{
   height: 5rem;
   width: 5rem;
   line-height: 5rem;
   font-size: 2rem;
   border:var(--border);
   color:#fff;
   margin:.3rem;
}
.products .box-container .box .icons a:hover{
   background:var(--main-color);
}
.products .box-container .box .image{
   padding: 2.5rem 0;
}
.products .box-container .box .image img{
   height: 20rem;
   width: 20rem;
   border-radius: 50%;
   border: 5px solid wheat;
}
.products .box-container .box .content h3{
   color:#fff;
   font-size: 2.5rem;
}
.products .box-container .box .content .stars{
   padding: 1.5rem;
}
.products .box-container .box .content .stars i{
   font-size: 1.7rem;
   color: var(--main-color);
}
.products .box-container .box .content .price{
   color:#fff;
   font-size: 2.5rem;
}
.products .box-container .box .content .price span{
   text-decoration: line-through;
   font-weight: lighter;
   font-size: 1.5rem;
}
.review .box-container{
   display: grid;
   grid-template-columns: repeat(auto-fit, minmax(10rem, 1fr));
   gap:1.0rem;
```

```
}
.review .box-container .box {
   text-align: center;
   padding:3rem 3rem;
}
.review .box-container .box p .flip-box{
   font-size: 1.5rem;
   line-height: 1.8;
   color:#ccc;
   padding:2rem 0;
}
.review .box-container .box .user{
   height: 7rem;
   width: 7rem;
   border-radius: 20%;
   object-fit: cover;
}
.contact .row{
   display: flex;
   background:var(--black);
   flex-wrap: wrap;
   gap:1rem;
   line-height: 1.8
}
.contact .row .map{
   flex:1 1 45rem;
   width: 100%;
   object-fit: cover;
}
.contact .row form{
   flex:1 1 45rem;
   padding:5rem 2rem;
   text-align: center;
   line-height: 1.8
}
.contact .row form h3{
   text-transform: uppercase;
   font-size: 3.5rem;
   color:#fff;
   line-height: 1.8
}
.contact .row form .inputBox{
   display: flex;
   align-items: center;
   margin-top: 2rem;
   margin-bottom: 2rem;
```

```
    background:var(--bg);
    border:var(--border);
    line-height: 1.8
}
.contact .row form .inputBox span{
    color:#fff;
    font-size: 2rem;
    padding-left: 2rem;
    line-height: 1.8
}
.contact .row form .inputBox input{
    width: 100%;
    padding:2rem;
    font-size: 1.7rem;
    color:#fff;
    text-transform: none;
    background:none;
    line-height: 1.8
}
.blogs .box-container{
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
    gap:1.5rem;
}
.blogs .box-container .box{
    border:var(--border);
}
.blogs .box-container .box .image{
    height: 25rem;
    overflow:hidden;
    width: 100%;
}
.blogs .box-container .box .image img{
    height: 100%;
    object-fit: cover;
    width: 100%;
}
.blogs .box-container .box:hover .image img{
    transform: scale(1.2);
}
.blogs .box-container .box .content{
    padding:2rem;
}
.blogs .box-container .box .content .title{
    font-size: 2.5rem;
    line-height: 1.5;
```

```
    color:#fff;
}
.blogs .box-container .box .content .title:hover{
    color:var(--main-color);
}
.blogs .box-container .box .content span{
    color:var(--main-color);
    display: block;
    padding-top: 1rem;
    font-size: 2rem;
}
.blogs .box-container .box .content p{
    font-size: 1.6rem;
    line-height: 1.8;
    color:#ccc;
    padding:1rem 0;
}
.footer{
    background:var(--black);
    text-align: center;
}
.footer .share{
    padding:1rem 0;
}
.footer .share a{
    height: 5rem;
    width: 5rem;
    line-height: 5rem;
    font-size: 2rem;
    color:#fff;
    border:var(--border);
    margin:.3rem;
    border-radius: 50%;
}
.footer .share a:hover{
    background-color: var(--main-color);
}
.footer .links{
    display: flex;
    justify-content: center;
    flex-wrap: wrap;
    padding:2rem 0;
    gap:1rem;
}
.footer .links a{
    padding:.7rem 2rem;
```

```
    color:#fff;
    border:var(--border);
    font-size: 2rem;
}
.footer .links a:hover{
    background:var(--main-color);
}
.footer .credit{
    font-size: 2rem;
    color:#fff;
    font-weight: lighter;
    padding:1.5rem;
}
.footer .credit span{
    color:var(--main-color);
}
/* media queries  */
@media (max-width:991px){
    html{
        font-size: 55%;
    }
    .header{
        padding:1.5rem 2rem;
    }
    section{
        padding:2rem;
}
@media (max-width:768px){
    #menu-btn{
        display: inline-block;
    }
    .header .navbar{
        position: absolute;
        top:100%; right: -100%;
        background: #fff;
        width: 30rem;
        height: calc(100vh - 9.5rem);
    }
    .header .navbar.active{
        right:0;
    }
    .header .navbar a{
        color:var(--black);
        display: block;
        margin:1.5rem;
        padding:.5rem;
```

```
        font-size: 2rem;
    }
    .header .search-form{
        width: 90%;
        right: 2rem;
    }
    .home{
        background-position: left;
        justify-content: center;
        text-align: center;
    }
    .home .content h3{
        font-size: 4.5rem;
    }
    .home .content p{
        font-size: 1.5rem;
    }
}
@media (max-width:450px){
    html{
        font-size: 50%;
    }
}
.notification-box {
    position: fixed;
    bottom: 1rem;
    right: 1rem;
    background-color: #fff;
    padding: 1rem;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
    z-index: 9999;
    display: none;
    transition: opacity 1s ease-out;
    border: 2px solid #ff0000;
}
.notification-box p {
    margin: 0;
    font-size: 1.6rem;
    color: #ff0000;
}
#ammoniaNotificationBox {
    bottom: 2rem;
}
#temperatureNotificationBox {
    bottom: 7rem;
```

```html
}
</style>
</head>
<body>
<header class="header">
   <a href="#" class="logo">
      <img src="../static/images/logo1.jpg" width="50" height="98" alt="">
   </a>
   <nav class="navbar">
      <a href="#home">Home</a>
      <a href="#about">About</a>
      <a href="#menu">Data Monitoring</a>
      <a href="#products">Preventions</a>
      <a href="#review">Team</a>
      <a href="#contact">Contact</a>
      <a href="/logout">Logout</a>
   </nav>
   <div class="icons">
      <div class="fas fa-bars" id="menu-btn"></div>
   </div>
</header>
<section class="home" id="home">
   <div class="content">
      <h3>GalluSpy Camera</h3>
      <p>Real-Time Monitoring System of Body Characteristics and Behavior of Gallus
Domesticus using Computer Vision</p>
   </div>
</section>
<section class="about" id="about">
   <h1 class="heading"> <span>about</span> us </h1>
   <div class="row";>
      <div class="image">
         <img src="../static/images/c2.jpg" alt="">
      </div>
      <div class="content">
         <h3>What is GalluSpy Camera?</h3>
         <p align = justify>GalluSpy Cam is a real-time camera monitoring system that
uses computer vision to monitor the behavior and body characteristics of the chickens
inside the poultry house. It features thermal imaging to detect the chickens' body
temperature, air quality sensor to monitor the air quality of the poultry house, and IoT
using Raspberry Pi to show real-time reports. </p>
         <p align = justify>The study aims to monitor the health of the chickens and detect
the possible diseases/disorders occurring in the poultry house.</p>
         <a href="#" class="btn">learn more</a>
      </div>
   </div>
```

```
</section>
<section class="menu" id="menu">
   <h1 class="heading"> Data <span>Monitoring</span> </h1>
   <div class="box-container">
      <div class="box">
         <img src="./static/images/w1n.png" width="86" height="86" alt="">
         <h3>Live Weight</h3>
         <div class="price"></div>
         <a href="/webcam" class="btn">View</a>
      </div>
      <div class="box">
         <img src="./static/images/tmp2.png" width="86" height="86" alt="">
         <h3>Live Body Temperature</h3>
         <div class="price"></div>
         <a href= "/indextemp1" class="btn">View</a>
      </div>
      <div class="box">
         <img src="./static/images/c3n.png" width="86" height="86" alt="">
         <h3>Behavior and Mortality Rate</h3>
         <div class="price"></div>
         <a href="/webcam2" class="btn">View</a>
      </div>
      <div class="box">
         <img src="./static/images/am2.png" width="86" height="86" alt="">
         <h3>Level of Ammonia</h3>
         <div class="price"></div>
         <a href="/mqchart" class="btn">View</a>
      </div>
   </div>
</section>
<section class="products" id="products">
   <h1 class="heading"> Diseases, Records <span> and Preventions</span> </h1>
   <div class="box-container">
      <div class="box">
         <div class="btn"
            <a href="/prevt" class="btn">View</a>
         </div>
         <div class="image">
            <img src="./static/images/cht.png" width="100" height="100" alt="">
         </div>
         <div class="content">
            <h3>Heat related Diseases</h3>
         </div>
      </div>
      <div class="box">
         <div class="btn">
```

```html
            <a href="/preva" class="btn">View</a>
          </div>
          <div class="image">
            <img src="./static/images/ammn1.png" width="100" height="100" alt="">
          </div>
          <div class="content">
            <h3>Ammonia Diseases</h3>
          </div>
        </div>
        <div class="box">
          <div class="btn">
            <a href="/data" class="btn">View</a>
          </div>
          <div class="image">
            <img src="./static/images/wrcd.png" width="100" height="100" alt=""
          </div>
          <div class="content">
            <h3>Records and History</h3>
          </div>
        </div>
        <div class="box">
          <div class="btn">
            <a href="/prevv" class="btn">View</a>
          </div>
          <div class="image">
            <img src="./static/images/VITA1.png" width="100" height="100" alt="">
          </div>
          <div class="content"
            <h3>Vitamins</h3>
          </div>
        </div>
      </div>
    </section>
    <section class="review" id="review">
      <h1 class="heading"> Team </h1>
      <div class="box-container">
        <div class="box">
          <div class="flip-box">
            <div class="flip-box-inner">
              <div class="flip-box-front">
                <img src="./static/images/echo.png" width="110" height="100" alt="">
              </div>
              <div class="flip-box-back">
              <div class = "center"> <h1>Jericho</h1> </div>
              </div>
            </div>
```

```
          </div>
        </div>
        <div class="box">
          <div class="flip-box">
            <div class="flip-box-inner">
              <div class="flip-box-front">
                <img src="./static/images/SAN JUAN.png" width="110" height="100"
alt="">
              </div>
              <div class="flip-box-back">
              <div class = "center"> <h1>Errol</h1> </div>
              </div>
            </div>
          </div>
        </div>
        <div class="box">
          <div class="flip-box">
            <div class="flip-box-inner">
              <div class="flip-box-front">
                <img src="./static/images/RUIZ.png" width="110" height="100" alt="">
              </div>
              <div class="flip-box-back">
              <div class = "center"> <h1>Vhince</h1> </div>
              </div>
            </div>
          </div>
        </div>
        <div class="box">
          <div class="flip-box">
            <div class="flip-box-inner">
              <div class="flip-box-front">
                <img src="./static/images/CATUDIO.png" width="110" height="100"
alt="">
              </div>
              <div class="flip-box-back">
              <div class = "center"> <h1>Anne</h1> </div>
              </div>
            </div>
          </div>
        </div>
         <div class="box">
          <div class="flip-box">
            <div class="flip-box-inner">
              <div class="flip-box-front">
                <img src="./static/images/BUBAN.png" width="110" height="100"
alt="">
```

```
                </div>
                <div class="flip-box-back">
                <div class = "center"> <h1>Zian</h1> </div>
                </div>
             </div>
          </div>
       </div>
    </div>
</section>
<section class="contact" id="contact">
    <h1 class="heading"> <span>contact</span> us </h1>
    <div class="row">
        <div class="mapouter"><div class="gmap_canvas"><iframe class="gmap_iframe" 
frameborder="0" scrolling="no" marginheight="0" marginwidth="0" 
src="https://maps.google.com/maps?width=324&amp;height=303&amp;hl=en&amp;q=
Technological University of the 
Philippines&amp;t=&amp;z=14&amp;ie=UTF8&amp;iwloc=B&amp;output=embed"></
iframe><a 
href="https://mcpenation.com/">https://mcpenation.com</a></div><style>.mapouter{po
sition:relative;text-align:right;width:324px;height:303px;}.gmap_canvas 
{overflow:hidden;background:none!important;width:324px;height:303px;}.gmap_iframe 
{width:324px!important;height:303px!important;}</style></div>
        <form action="">
           <div class="image">
              <img 
src="./static/images/Technological_University_of_the_Philippines_Seal.svg.png" 
width="90" height="90" alt="">
           </div>
           <div class="content">
              <font size="3" face="arial" color="white">
           <p align = center> Technological University of the Philippines </p>
           </div>
           <div class="image">
              <img src="./static/images/coe3.png" width="90" height="90" alt="">
           </div>
           <div class="content">
              <font size="3" face="arial" color="white">
           <p align = center> College of Engineering </p>
           </div>
        </form>
    </div>
</section>
<section class="footer">
    <div class="share">
        <a href="#" class="fab fa-facebook-f"></a>
        <a href="#" class="fab fa-twitter"></a>
```

```
        <a href="#" class="fab fa-instagram"></a>
        <a href="#" class="fab fa-linkedin"></a>
        <a href="#" class="fab fa-pinterest"></a>
    </div>
    <div class="links">
        <a href="#">home</a>
        <a href="#">About</a>
        <a href="#">Data Monitoring</a>
        <a href="#">Preventions</a>
        <a href="#">Team</a>
        <a href="#">contact</a>
    </div>
    <div class="credit">created by <span> Flock N' Flare'</span> | all rights
reserved</div>
</section>
    <div class="notification-box" id="ammoniaNotificationBox">
        <p>Warning: Ammonia level reached 30 PPM!</p>
    </div>
    <div class="notification-box" id="temperatureNotificationBox">
        <p>Warning: Temperature reached 40 Degrees!</p>
    </div>
    <script>
        function showNotificationBox(elementId) {
            var notificationBox = document.getElementById(elementId);
            notificationBox.style.display = 'block';
            setTimeout(function() {
                hideNotificationBox(elementId);
            }, 5000);
        }
        function hideNotificationBox(elementId) {
            var notificationBox = document.getElementById(elementId);
            notificationBox.style.opacity = '0'; // Fade-out effect
            setTimeout(function() {
                notificationBox.style.display = 'none';
                notificationBox.style.opacity = '1';
            }, 1000);
        }
        // Check ammonia level and show notification if necessary
        function checkAmmoniaLevel() {
            var xhr = new XMLHttpRequest();
            xhr.open('GET', '/get_ammonia', true);
            xhr.onreadystatechange = function() {
                if (xhr.readyState === XMLHttpRequest.DONE) {
                    if (xhr.status === 200) {
                        var ammoniaData = JSON.parse(xhr.responseText);
                        var ammoniaLevel = ammoniaData[ammoniaData.length - 1].ammonia;
```

```
            if (ammoniaLevel >= 24) {
              showNotificationBox('ammoniaNotificationBox');
            }
          } else {
            console.error('Error: ' + xhr.status);
          }
        }
      };
      xhr.send();
    }
    // Check temperature and show notification if necessary
    function checkTemperature() {
      var xhr = new XMLHttpRequest();
      xhr.open('GET', '/get_temperature', true);
      xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE) {
          if (xhr.status === 200) {
            var temperatureData = JSON.parse(xhr.responseText);
            var temperature = temperatureData[temperatureData.length - 1].temperature
            if (temperature >= 33) {
              showNotificationBox('temperatureNotificationBox');
            }
          } else {
            console.error('Error: ' + xhr.status);
          }
        }
      };
      xhr.send();
    }
    window.addEventListener('load', function() {
      setInterval(function() {
        checkAmmoniaLevel();
        setTimeout(checkTemperature, 5000);
      }, 5000);
    window.addEventListener('load', function() {
      setInterval(function() {
        checkTemperature();
        setTimeout(checkAmmoniaLevel, 5000);
      }, 5000);
    });
  </script>
</body>
</html>
```

charts.html
<!—Live Temperature Graph and Parameters -->

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Sensors Chart</title>
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
  <style>
    body {
      background: url('../static/images/bg1d.png') no-repeat center center fixed;
      background-size: cover;
      color: white; /* Change font color to white */
      font-family: 'Roboto', sans-serif;
      margin: 0;
      padding: 0;
    }
    h1 {
      text-align: center;
      font-size: 30px;
      margin-top: 20px;
    }
    #graphDiv {
      margin: 50px auto;
      width: 80%;
      height: 400px;
    }
    #warningDiv {
      display: none;
      margin-top: 20px;
      text-align: center;
      color: white;
      background-color: rgba(255, 0, 0, 0.6);
      padding: 10px;
    }
    #parametersBox {
      position: absolute;
      top: 50px;
      right: 50px;
      background-color: wheat; /* Change background color to wheat */
      padding: 10px;
      font-size: 16px;
      color: black; /* Change font color to black */
      border: 1px solid black; /* Add black border */
    }
  </style>
</head>
<body>
```

```
<h1>Temperature Monitoring <span style="font-size: 18px"></span></h1>
<div id="graphDiv"></div>
<div id="warningDiv"></div>
<div id="parametersBox"></div>
<script>
  var xValues = [];
  var yValues = [];
  var layout = {
    title: {
      text: 'Live Temperature Graph',
      font: {
        color: '#fff'
      }
    },
    xaxis: {
      title: {
        text: 'Time',
        font: {
          color: '#fff'
        }
      },
      tickfont: {
        color: '#fff'
      }
    },
    yaxis: {
      title: {
        text: 'Temperature Level',
        font: {
          color: '#fff'
        }
      },
      tickfont: {
        color: '#fff'
      }
    },
    paper_bgcolor: 'transparent',
    plot_bgcolor: 'transparent'
  };
  var trace = {
    x: xValues,
    y: yValues,
    type: 'line',
    line: { color: 'rgba(255, 0, 0, 0.6)' }
  };
  var graphDiv = document.getElementById('graphDiv');
```

```
        var chartData = [trace];
        Plotly.newPlot(graphDiv, chartData, layout);
        function showWarningMessage(message) {
          var warningDiv = document.getElementById('warningDiv');
          warningDiv.innerHTML = message;
          warningDiv.style.display = 'block';
        }
        function hideWarningMessage() {
          var warningDiv = document.getElementById('warningDiv');
          warningDiv.style.display = 'none';
        }
        function updateGraph() {
          fetch('/get_temperature')
            .then(response => response.json())
            .then(data => {
              xValues = [];
              yValues = [];
              data.forEach(item => {
                xValues.push(item.time);
                yValues.push(item.temperature[0].toFixed(3));
              });
              Plotly.update(graphDiv, { x: [xValues], y: [yValues] });
              // Get the latest temperature value
              var latestTemperature = yValues[yValues.length - 1];
              // Check temperature levels and show warning messages
              if (latestTemperature >= 30 && latestTemperature < 35) {
                showWarningMessage('Temperature level is between 30-35 degrees.
Chickens constantly beak gape and spread wings.');
              } else if (latestTemperature >= 35 && latestTemperature < 37) {
                showWarningMessage('Temperature level is between 35-37 degrees.
Chickens frequently beak gape and spread wings.');
              } else if (latestTemperature >= 37 && latestTemperature < 39) {
                showWarningMessage('Temperature level is between 37-39 degrees.
Chickens regularly beak gape and spread wings.');
              } else if (latestTemperature >= 39 && latestTemperature < 41) {
                showWarningMessage('Temperature level is between 39-41 degrees.
Chickens occasionally beak gape and spread wings.');
              } else if (latestTemperature >= 41 && latestTemperature < 45) {
                showWarningMessage('Temperature level is between 41-45 degrees.
Chickens rarely beak gape and spread wings.');
              } else {
                hideWarningMessage();
              }
              // Update parameters box with latest data
              var latestTime = xValues[xValues.length - 1];
              var latestDate = new Date().toLocaleDateString();
```

```
                var parametersBox = document.getElementById('parametersBox');
                parametersBox.innerHTML = '<strong>Latest Temperature:</strong> ' +
latestTemperature + '°C<br>' +
                                          '<strong>Date:</strong> ' + latestDate + '<br>' +
                                          '<strong>Time:</strong> ' + latestTime;
            });
        }
        setInterval(updateGraph, 1000);
    </script>
</body>
</html>
```

mqcharts.html
<!—Live Ammonia Graph and Parameters -->

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Sensors Chart</title>
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
  <style>
     body {
        background: url('../static/images/bg1d.png') no-repeat center center fixed;
        background-size: cover;
        color: white; /* Change font color to white */
        font-family: 'Roboto', sans-serif;
        margin: 0;
        padding: 0;
     }
     h1 {
        text-align: center;
        font-size: 30px;
        margin-top: 20px;
     }
     #graphDiv {
        margin: 50px auto;
        width: 80%;
        height: 400px;
        background-color: rgba(255, 255, 255, 0.5);
     }
     .warning-message {
        text-align: center;
        color: white;
        font-weight: bold;
        font-size: 20px;
        visibility: hidden;
```

```
            background-color: rgba(0, 0, 0, 0.5); /* Add background color */
            padding: 10px; /* Add padding for the box */
        }
        #parametersBox {
            position: absolute;
            top: 50px;
            right: 50px;
            background-color: wheat; /* Change background color to wheat */
            padding: 10px;
            font-size: 16px;
            color: black; /* Change font color to black */
            border: 1px solid black; /* Add black border */
        }
    </style>
</head>
<body>
    <h1>Ammonia Monitoring <span style="font-size: 18px"></span></h1>
    <div id="graphDiv"></div>
    <div id="warningMessage" class="warning-message"></div>
    <div id="parametersBox"></div>
    <script>
        document.addEventListener('DOMContentLoaded', function() {
            var xValues = [];
            var yValues = [];
            var graphDiv = document.getElementById('graphDiv');
            var warningDiv = document.getElementById('warningMessage');
            var parametersBox = document.getElementById('parametersBox');
            var chartData = [{
                x: xValues,
                y: yValues,
                type: 'line',
                line: { color: 'blue' }
            }];
            var layout = {
                title: {
                    text: 'Live Ammonia Level Graph',
                    font: {
                        color: '#fff'
                    }
                },
                xaxis: {
                    title: {
                        text: 'Time',
                        font: {
                            color: '#fff'
                        }
```

```
        },
        tickfont: {
            color: '#fff'
        }
    },
    yaxis: {
        title: {
            text: 'Ammonia Level',
            font: {
                color: '#fff'
            }
        },
        tickfont: {
            color: '#fff'
        }
    },
    paper_bgcolor: 'transparent',
    plot_bgcolor: 'transparent'
};
function updateGraph() {
    fetch('/get_ammonia')
        .then(response => response.json())
        .then(data => {
            xValues = [];
            yValues = [];
            data.forEach(item => {
                var datetimeParts = item.time.split(':');
                var datetime = new Date();
                datetime.setHours(datetimeParts[0]);
                datetime.setMinutes(datetimeParts[1]);
                datetime.setSeconds(datetimeParts[2]);
                datetime.setMilliseconds(0);
                xValues.push(datetime);
                yValues.push(item.ammonia);
            });
            Plotly.update(graphDiv, { x: [xValues], y: [yValues] }, layout);
            updateWarningMessage();
            updateParametersBox(data);
        })
        .catch(error => {
            console.error('Error fetching ammonia data:', error);
            hideWarningMessage();
        });
}
function updateWarningMessage() {
    if (yValues.length > 0) {
```

```
            var ammoniaLevel = yValues[yValues.length - 1];
            var message = '';
            if (ammoniaLevel >= 30 && ammoniaLevel < 35) {
              message = 'Ammonia level is between 30-35 PPM. Chickens constantly
scratch and shake their heads.';
            } else if (ammoniaLevel >= 25 && ammoniaLevel < 30) {
              message = 'Ammonia level is between 25-30 PPM. Chickens frequently
scratch and shake their heads.';
            } else if (ammoniaLevel >= 20 && ammoniaLevel < 25) {
              message = 'Ammonia level is between 20-25 PPM. Chickens regularly
scratch and shake their heads.';
            } else if (ammoniaLevel >= 15 && ammoniaLevel < 20) {
              message = 'Ammonia level is between 15-20 PPM. Chickens occasionally
scratch and shake their heads.';
            } else if (ammoniaLevel >= 10 && ammoniaLevel < 15) {
              message = 'Ammonia level is between 10-15 PPM. Chickens rarely
scratch and shake their heads.';
            }
            if (message) {
              showWarningMessage(message);
            } else {
              hideWarningMessage();
            }
          } else {
            hideWarningMessage();
          }
        }
        function showWarningMessage(message) {
          warningDiv.innerHTML = message;
          warningDiv.style.visibility = 'visible';
        }
        function hideWarningMessage() {
          warningDiv.style.visibility = 'hidden';
        }
        function updateParametersBox(data) {
          if (data.length > 0) {
            var latestData = data[data.length - 1];
            var ammoniaFormatted = latestData.ammonia.toFixed(1);
            var parametersHTML = `
              <strong>Ammonia Level:</strong> ${ammoniaFormatted} PPM<br>
              <strong>Date:</strong> ${latestData.date || ''}<br>
              <strong>Time:</strong> ${latestData.time || ''}
            `;
            parametersBox.innerHTML = parametersHTML;
          }
        }
```

```
            Plotly.newPlot(graphDiv, chartData, layout);
            setInterval(updateGraph, 5000);
        });
    </script>
</body>
</html>
```

ui.html

<!—Live Weight Monitoring of Chicken -->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no,
initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Live Monitoring of Chicken Behavior and Mortality</title>
    <style>
        body {
            color: white;
            margin: 0px;
            padding: 0px;
            background-color: black;
            font-family: Calibri, sans-serif;
            height: 0;
            width: 100%;
            justify-content: center;
        }
        header.feature-box.top {
            background-color: black;
            height: 50px;
            margin: 0px;
            padding: 20px;
            text-align: center;
        }
        header.feature-box.second {
            background-color: black;
            height: 50px;
            text-align: center;
            margin-top: -25px;
        }
        .features {
            background-color: black;
            width: 900px;
            height: 100px;
            border-radius: 35px;
```

```
    object-fit: contain;
    margin: 20px;
}
.col-sm {
    width: 50%;
    height: auto;
    margin: 0 auto
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
section.col-sm {
    background-color: black;
    width: 50%;
    border-radius: 1px;
    object-fit: contain;
    margin: 5px;
    overflow: auto;
}
.form-container {
    background-color: black;
    width: 300px;
    height: 300px;
    border-radius: 5px;
    object-fit: contain;
    margin: 40px;
    margin-top: 100px;
    padding: 1px;
    align-items: center;
    display: flex;
    justify-content: center;
}
form {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
form label {
    color: white;
    margin: 10px;
}
form input {
    padding: 10px;
```

```
  border-radius: 5px;
  border: none;
  margin: 5px;
}
form input[type=submit] {
  background-color: blue;
  color: white;
  cursor: pointer;
}
form input[type=submit]:hover {
  background-color: white;
  color: blue;
}
img {
  width: 100%;
  height: auto;
  border-radius: 3px;
  object-fit: contain;
  margin: auto;
  display: block;
}
table {
  width: 100%;
  height: 100px;
  overflow-y: scroll;
}
th,
td {
  padding: 10px;
  text-align: left;
}
th {
  background-color: blue;
  color: white;
}
.activity-container {
  width: 50%;
  height: 300px;
  padding: 10px;
  overflow-y: scroll;
  border: 10px solid #ccc;
  border-radius: 20px;
  align-items: center;
  margin-top: 50px;
  margin: 100px;
}
```

```
.activity-table {
   width: 100%;
   border-collapse: collapse;
   margin-bottom: 0;
}
.activity-table th,
.activity-table td {
   padding: 5px;
   text-align: center;
   vertical-align: middle;
   font-size: 20px;
   line-height: 1;
}
.activity-table th {
   background-color: black;
   font-weight: bold;
}
.recorded-activity-container {
   line-height: 0;
}
.videos-container {
   display: flex;
   justify-content: center;
   margin: 20px;
}
.box {
   position: relative;
   width: 200px;
   height: 150px;
   background-color: white;
   border-radius: 5px;
   margin: 20px;
   overflow: hidden;
}
.btn {
   position: absolute;
   width: 100%;
   height: 100%;
   top: 0;
   left: 0;
   display: flex;
   align-items: center;
   justify-content: center;
   text-decoration: none;
   background-color: white;
   opacity: 0;
```

```
            transition: opacity 0.3s ease;
            z-index: 1;
            pointer-events: none;
        }
        .btn:hover {
            opacity: 1;
        }
        .btn a {
            color: blue;
            font-size: 18px;
            font-weight: bold;
            text-transform: uppercase;
            font-family: "Calibri", sans-serif;
            pointer-events: auto;
        }
        .content {
            position: absolute;
            bottom: 0;
            left: 0;
            padding: 10px;
        }
        .content h3 {
            color: black;
            margin: 0;
            font-size: 24px;
            font-weight: bold;
            text-align: center;
            font-family: "Roboto", sans-serif;
        }
    </style>
</head>
<body>
    <header class="feature-box top">
        <h1><strong>Live Weight Monitoring of Chicken</strong></h1>
    </header>
    <header class="feature-box second">
        <h1><strong>Video Live Stream</strong></h1>
    </header>
    <div class="videos-container">
        <section class="col-sm">
            <img src="{{ url_for('webapp3') }}" alt="Web App 3 Video">
            <div class="box">
                <div class="btn">
                    <a href="{{ url_for('webapp4') }}" class="btn"></a>
                </div>
                <div class="content">
```

```
          <h3>Graph of Live Weight Monitoring of Chicken</h3>
        </div>
      </div>
    </section>
  </div>
</body>
</html>
```

ui2.hmtl

<!—Live Mortality Rate of Chicken -->

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, user-scalable=no,
initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Mortality Rate of Chicken</title>
  <style>
    body {
      color: white;
      margin: 0px;
      padding: 0px;
      background-color: black;
      font-family: Calibri, sans-serif;
      height: 0;
      width: 100%;
      justify-content: center;
    }
    header.feature-box.top {
      background-color: black;
      height: 50px;
      margin: 0px;
      padding: 20px;
      text-align: center;
    }
    header.feature-box.second {
      background-color: black;
      height: 50px;
      text-align: center;
      margin-top: -25px;
    }
    .features {
      background-color: black;
      width: 900px;
      height: 100px;
      border-radius: 35px;
```

```
    object-fit: contain;
    margin: 20px;
}
.col-sm {
    width: 50%;
    height: auto;
    margin: 0 auto
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
section.col-sm {
    background-color: black;
    width: 50%;
    border-radius: 1px;
    object-fit: contain;
    margin: 5px;
    overflow: auto;
}
.form-container {
    background-color: black;
    width: 300px;
    height: 300px;
    border-radius: 5px;
    object-fit: contain;
    margin: 40px;
    margin-top: 100px;
    padding: 1px;
    align-items: center;
    display: flex;
    justify-content: center;
}
form {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
form label {
    color: white;
    margin: 10px;
}
form input {
    padding: 10px;
```

```
  border-radius: 5px;
  border: none;
  margin: 5px;
}
form input[type=submit] {
  background-color: blue;
  color: white;
  cursor: pointer;
}
form input[type=submit]:hover {
  background-color: white;
  color: blue;
}
img {
  width: 100%;
  height: auto;
  border-radius: 3px;
  object-fit: contain;
  margin: auto;
  display: block;
}
table {
  width: 100%;
  height: 100px;
  overflow-y: scroll;
}
th,
td {
  padding: 10px;
  text-align: left;
}
th {
  background-color: blue;
  color: white;
}
.activity-container {
  width: 50%;
  height: 300px;
  padding: 10px;
  overflow-y: scroll;
  border: 10px solid #ccc;
  border-radius: 20px;
  align-items: center;
  margin-top: 50px;
  margin: 100px;
}
```

```
.activity-table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 0;
}
.activity-table th,
.activity-table td {
  padding: 5px;
  text-align: center;
  vertical-align: middle;
  font-size: 20px;
  line-height: 1;
}
.activity-table th {
  background-color: black;
  font-weight: bold;
}
.recorded-activity-container {
  line-height: 0;
}
.videos-container {
  display: flex;
  justify-content: center;
  margin: 20px;
}
.box {
  position: relative;
  width: 200px;
  height: 150px;
  background-color: white;
  border-radius: 5px;
  margin: 20px;
  overflow: hidden;
}
.btn {
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  display: flex;
  align-items: center;
  justify-content: center;
  text-decoration: none;
  background-color: white;
  opacity: 0;
```

```
            transition: opacity 0.3s ease;
            z-index: 1;
            pointer-events: none;
        }
        .btn:hover {
            opacity: 1;
        }
        .btn a {
            color: blue;
            font-size: 18px;
            font-weight: bold;
            text-transform: uppercase;
            font-family: "Calibri", sans-serif;
            pointer-events: auto;
        }
        .content {
            position: absolute;
            bottom: 0;
            left: 0;
            padding: 10px;
        }
        .content h3 {
            color: black;
            margin: 0;
            font-size: 24px;
            font-weight: bold;
            text-align: center;
            font-family: "Roboto", sans-serif;
        }
    </style>
</head>
<body>
    <header class="feature-box top">
        <h1><strong>Mortality Rate of Chicken</strong></h1>
    </header>
    <header class="feature-box second">
        <h1><strong>Video Live Stream</strong></h1>
    </header>
    <div class="videos-container">
        <section class="col-sm">
            <img src="{{ url_for('webapp') }}" alt="Web App 3 Video">
            <div class="box">
                <div class="btn">
                    <a href="{{ url_for('webapp2') }}" class="btn"></a>
                </div>
                <div class="content">
```

```
                <h3>Graph of Mortality Rate of Chicken</h3>
            </div>
        </div>
    </section>
</div>
</body>
</html>
```

ui3.html
<!—Live Graph Weight Monitoring of Chicken -->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no,
initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Live Weight Monitoring of Chicken</title>
    <style>
        body {
            color: white;
            margin: 0px;
            padding: 0px;
            background-color: black;
            font-family: Calibri, sans-serif;
            height: 0;
            width: 100%;
            justify-content: center;
        }
        header.feature-box.top {
            background-color: black;
            height: 50px;
            margin: 0px;
            padding: 20px;
            text-align: center;
        }
        header.feature-box.second {
            background-color: black;
            height: 50px;
            text-align: center;
            margin-top: -25px;
        }
        .features {
            background-color: black;
            width: 900px;
            height: 100px;
```

```
    border-radius: 35px;
    object-fit: contain;
    margin: 20px;
}
.col-sm {
    width: 50%;
    height: auto;
    margin: 0px;
    padding: 0px;
    display: inline-block;
    vertical-align: top;
}
section.col-sm {
    background-color: black;
    width: 100%;
    border-radius: 1px;
    object-fit: contain;
    margin: 5px;
    overflow: auto;
}
.form-container {
    background-color: black;
    width: 300px;
    height: 300px;
    border-radius: 5px;
    object-fit: contain;
    margin: 40px;
    margin-top: 100px;
    padding: 1px;
    align-items: center;
    display: flex;
    justify-content: center;
}
form {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
form label {
    color: white;
    margin: 10px;
}
form input {
    padding: 10px;
    border-radius: 5px;
```

```
    border: none;
    margin: 5px;
}
form input[type=submit] {
    background-color: blue;
    color: white;
    cursor: pointer;
}
form input[type=submit]:hover {
    background-color: white;
    color: blue;
}
img {
    width: 100%;
    height: auto;
    border-radius: 3px;
    object-fit: contain;
    margin: auto;
    display: block;
}
table {
    width: 100%;
    height: 100px;
    overflow-y: scroll;
}
th,
td {
    padding: 10px;
    text-align: left;
}
th {
    background-color: blue;
    color: white;
}
.activity-container {
    width: 50%;
    height: 300px;
    padding: 10px;
    overflow-y: scroll;
    border: 10px solid #ccc;
    border-radius: 20px;
    align-items: center;
    margin-top: 50px;
    margin: 100px;
}
.activity-table {
```

```
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 0;
        }
        .activity-table th,
        .activity-table td {
            padding: 5px;
            text-align: center;
            vertical-align: middle;
            font-size: 20px;
            line-height: 1;
        }
        .activity-table th {
            background-color: black;
            font-weight: bold;
        }
        .recorded-activity-container {
            line-height: 0;
        }
        .videos-container {
            display: flex;
            justify-content: space-between;
            margin: 20px;
        }
    </style>
</head>
<body>
    <header class="feature-box top">
        <h1><strong>Live Weight Monitoring of Chicken</strong></h1>
    </header>
    <header class="feature-box second">
        <h1><strong>Video Live Stream</strong></h1>
    </header>
    <div class="videos-container">
        <section class="col-sm">
            <img src="{{ url_for('webapp4') }}" alt="Web App 3 Video">
        </section>
    </div>
</body>
</html>
```

ui4.html

<!—Live Graph Mortality Rate of Chicken -->

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no,
initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Live Monitoring of Chicken Behavior and Mortality</title>
    <style>
        body {
            color: white;
            margin: 0px;
            padding: 0px;
            background-color: black;
            font-family: Calibri, sans-serif;
            height: 0;
            width: 100%;
            justify-content: center;
        }
        header.feature-box.top {
            background-color: black;
            height: 50px;
            margin: 0px;
            padding: 20px;
            text-align: center;
        }
        header.feature-box.second {
            background-color: black;
            height: 50px;
            text-align: center;
            margin-top: -25px;
        }
        .features {
            background-color: black;
            width: 900px;
            height: 100px;
            border-radius: 35px;
            object-fit: contain;
            margin: 20px;
        }
        .col-sm {
            width: 50%;
            height: auto;
            margin: 0px;
            padding: 0px;
            display: inline-block;
            vertical-align: top;
        }
        section.col-sm {
```

```
    background-color: black;
    width: 100%;
    border-radius: 1px;
    object-fit: contain;
    margin: 5px;
    overflow: auto;
}
.form-container {
    background-color: black;
    width: 300px;
    height: 300px;
    border-radius: 5px;
    object-fit: contain;
    margin: 40px;
    margin-top: 100px;
    padding: 1px;
    align-items: center;
    display: flex;
    justify-content: center;
}
form {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
form label {
    color: white;
    margin: 10px;
}
form input {
    padding: 10px;
    border-radius: 5px;
    border: none;
    margin: 5px;
}
form input[type=submit] {
    background-color: blue;
    color: white;
    cursor: pointer;
}
form input[type=submit]:hover {
    background-color: white;
    color: blue;
}
img {
```

```
  width: 100%;
  height: auto;
  border-radius: 3px;
  object-fit: contain;
  margin: auto;
  display: block;
}
table {
  width: 100%;
  height: 100px;
  overflow-y: scroll;
}
th,
td {
  padding: 10px;
  text-align: left;
}
th {
  background-color: blue;
  color: white;
}
.activity-container {
  width: 50%;
  height: 300px;
  padding: 10px;
  overflow-y: scroll;
  border: 10px solid #ccc;
  border-radius: 20px;
  align-items: center;
  margin-top: 50px;
  margin: 100px;
}
.activity-table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 0;
}
.activity-table th,
.activity-table td {
  padding: 5px;
  text-align: center;
  vertical-align: middle;
  font-size: 20px;
  line-height: 1;
}
.activity-table th {
```

```
            background-color: black;
            font-weight: bold;
        }
        .recorded-activity-container {
            line-height: 0;
        }
        .videos-container {
            display: flex;
            justify-content: space-between;
            margin: 20px;
        }
    </style>
</head>
<body>
    <header class="feature-box top">
        <h1><strong>Live Weight Monitoring of Chicken</strong></h1>
    </header>
    <header class="feature-box second">
        <h1><strong>Video Live Stream</strong></h1>
    </header>
    <div class="videos-container">
        <section class="col-sm">
            <img src="{{ url_for('webapp2') }}" alt="Web App 3 Video">
        </section>
    </div>
</body>
</html>
```

ANNEX B

Evaluation Form

Respondents Informations

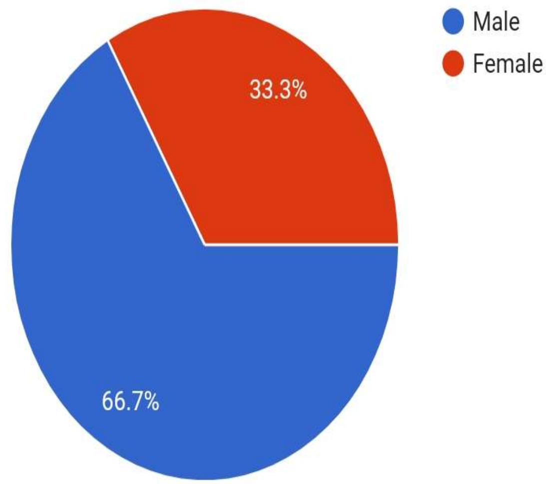
Address/Location

6 responses

- Gasan, Marinduque
- Purok 4 Tipolo Ubay Bohol
- Purok 6, Sta. Monica, Floridablanca, Pampanga
- Alalum, San Pascual, Batangas
- Guagua, Pampanga
- 15 kabuyao st. Bilog Balangkas Valenzuela City

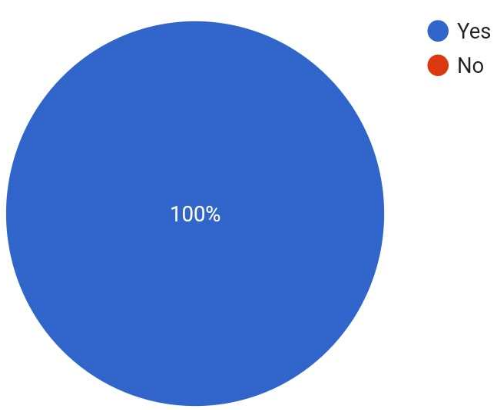
Gender

6 responses



Are you a poultry owner? (Nagmamayari ba kayo ng manukan o nag-aalaga ba kayo ng mga manok?)

6 responses



Years of being a poultry owner (Taon ng pagiging may-ari ng manok)

6 responses

- 2
- 5years
- 4
- 4yrs
- 34
- 30 years

RESPONDENT

I, at this moment, authorize the organizers to collect and process the data indicated herein to improve the organization's future events. I understand that my personal information is protected by RA 10173, Data Privacy Act of 2012.

- (●) Yes. I do agree and accept the Data Privacy Agreement

- (●) Yes. I do agree and accept the Non-Disclosure Agreement

PERSONAL INFORMATION

Full Name (Juan M. Dela Cruz) *

Olivia San Juan

Address/Location *

Gasan, Marinduque

Gender *

- (●) Male
- ( ) Female

Are you a poultry owner? (Nagmamayari ba kayo ng manukan o nag-aalaga ba kayo ng mga manok?) *

- (●) Yes
- ( ) No

Years of being a poultry owner (Taon ng pagiging may-ari ng manok) *

2

Summary of Responses from Evaluation Form

Statements	Rating				
	1	2	3	4	5
Hardware and Software					
The cable management of the prototype is well arranged.			2	1	3
The components used in the prototype are compatible.			1	3	2
The interface of the website can be accessed easily.			2	2	2
The information listed in the web application is informative and reliable.				3	3
Output					
The system can identify the behavior and body characteristics of the chicken and the room temperature of the coop.			1	2	3
The data displayed are accurate and can easily be understood.			2	2	2
The preventions are relevant to answer the problem.			1	1	4
Functionality and Efficiency					
The monitoring system is sensitive enough to determine the behavior and body characteristics of the chickens.			3	2	1
The system gathers information more efficiently than manual monitoring.			4	1	1
The system is not easily disrupted by any minimal physical disturbance.			3	1	2
The prototype is cost-efficient.			2	3	1
Operability					
The data is transmitted with minimal delay.			1	3	2
The website shows the data gathered correctly.			2	3	1
The web application is user-friendly.			2	2	2

ANNEX C

Device Specification Sheets

Microsoft®
LifeCam HD-3000

Name Information	
Product Name	Microsoft® LifeCam HD-3000
Webcam Name	Microsoft LifeCam HD-3000
Product Dimensions	
Webcam Length	1.55 inches (39.3 millimeters)
Webcam Width	1.75 inches (44.5 millimeters)
Webcam Depth/Height	4.28 inches (109 millimeters)
Webcam Weight	3.17 ounces (89.9 grams)
Webcam Cable Length	59.1 inches (1500 millimeters)
Compatibility and Localization	
Interface	High-speed USB compatible with the USB 2.0 specification
Operating Systems	Microsoft Windows®® 10 / 8.1 / 8 / RT 8.1 / RT 8 / Windows 7, Windows Vista®, and Windows XP with Service Pack 2 (SP2) excluding Windows XP Pro 64-bit
Top-line System Requirements	Requires a PC that meets the requirements for and has installed one of these operating systems: • Microsoft Windows 10 / 8.1 / 8 / RT 8.1 / RT 8 / Windows 7, Windows Vista, and Windows XP with Service Pack 2 (SP2) excluding Windows XP Pro 64-bit **For VGA video calling:** • Intel Dual Core 1.6 GHz or higher • 1 GB of RAM **For 720p HD recording:** • Intel Dual Core 3.0 GHz or higher • 2 GB of RAM • 1.5 GB of hard drive space • USB 2.0 required • Windows-compatible speakers or headphones You must accept License Terms for software download. Please download the latest available software version for your OS/Hardware combination Internet access may be required for certain features. Local and/or long-distance telephone toll charges may apply. Software download required for full functionality of all features. Internet functions (post to Windows Live™ Spaces, send in e-mail, video calls), also require: Internet Explorer® 6/7/8 browser software required for installation; 25 MB hard drive space typically required (users can maintain other default Web browsers after installation)
Compatibility Logos	• Compatible with Microsoft Windows 10 / 8 / 7 • Skype™ for Business Certified
Software Localization	Microsoft LifeCam software version 3.0 may be installed in Simplified Chinese, Traditional Chinese, English, French, German, Italian, Japanese, Korean, Brazilian Portuguese, Iberian Portuguese, Russian, or Spanish. If available, standard setup will install the software in the default OS language. Otherwise, the English language version will be installed.
Windows Live™ Integration Features	
Video Conversation Feature	Windows Live call button delivers one touch access to video conversation
Call Button Life	10,000 actuations
Webcam Controls & Effects	LifeCam Dashboard provides access to animated video effects and webcam controls
Windows Live Integration Features	Windows Live Photo Gallery integration - Take a photo with LifeCam Software, then with one click open Photo Gallery to edit, tag and share it online Windows Live Movie Maker integration - Record a video with LifeCam Software and start a movie project on Movie Maker with just one click to then upload it to your favorite networking site
Imaging Features	
Sensor	CMOS sensor technology
Resolution[1]	• Motion Video: 1280 X 720 pixel resolution • Still Image: 1280 X 800 pixel resolution = 1,024,000 pixels [1] One megapixel = 1,000,000 pixels. Lower resolution available when sending video via instant messaging.
Imaging Rate	Up to 30 frames per second
Field of View	68.5° diagonal field of view
Imaging Features	• Digital pan, digital tilt, vertical tilt, swivel pan, and 4x digital zoom • Fixed focus from 0.3m to 1.5m • True Color - Automatic image adjustment with manual override • 16:9 widescreen • 24-bit color depth
Product Feature Performance	
Audio Features	Integrated microphone
Microphone Technology	Omni directional microphone
Frequency Range	Frequency range 200Hz – 20kHz
Mounting Features	Flexible universal attachment base
Storage Temperature & Humidity	-40 °F (-40 °C) to 140 °F (60 °C) at <5% to 65% relative humidity (non-condensing)
Operating Temperature & Humidity	32° F (0° C) to 104 °F (40 °C) at <5% to 80% relative humidity (non-condensing)
Certification Information	
Country of Manufacture	People's Republic of China
ISO 9001 Qualified Manufacturer	Yes
ISO 14001 Qualified Manufacturer	Yes
Restriction on Hazardous Substances	This device complies with all applicable worldwide regulations and restrictions including, but not limited to: EU directive 2002/95/EC on the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment and EU Registration Evaluation and Authorization of Chemicals (REACH) regulation regarding Substances of Very High Concern.
FCC ID	This device complies with part 15 of the FCC Rules and Industry Canada ICES-003. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Tested to comply with FCC standards. For home and office use. Model number: 1492, LifeCam HD-3000.
Agency and Regulatory Marks	• ACMA Declaration of Conformity (Australia and New Zealand) • ICES-003 report on file (Canada) • EIP Pollution Control Mark, EPUP (China) • CE Declaration of Conformity (European Union) • WEEE (European Union) • VCCI Certificate (Japan) • KCC Certificate (Korea) • EAC Certificate (Russia, Kazakhstan) • FCC Declaration of Conformity (USA) • CB Scheme Certificate (International)
Warranty	3 years
Windows Certification Kit (WCK)	ID: 1751964 (32-bit and 64-bit) Microsoft Windows 10, 1608508 (32-bit) and 1608509 (64-bit) Microsoft Windows 8.1

TECHNICAL DATA MQ-135 GAS SENSOR

FEATURES

Wide detecting scope   Fast response and High sensitivity
Stable and long life   Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH3,NOx, alcohol, Benzene, smoke,CO_2 ,etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33 Ω ±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃-45℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Ramark 2
Rs	Sensing Resistance	30KΩ-200KΩ (100ppm NH_3)	Detecting concentration scope: 10ppm-300ppm NH_3 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH_3	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20℃±2℃ Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

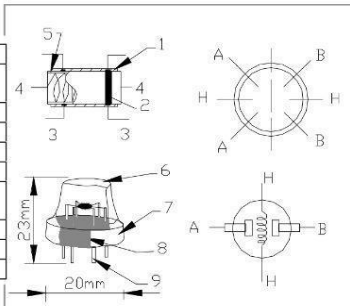


Fig. 1

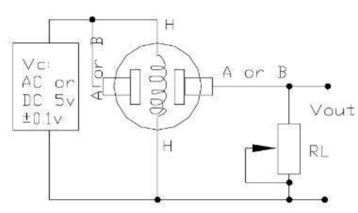
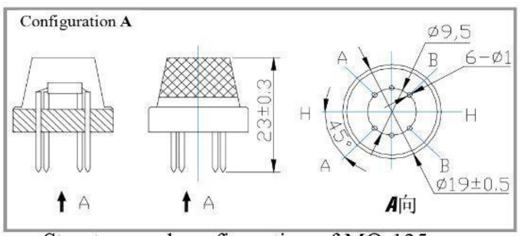
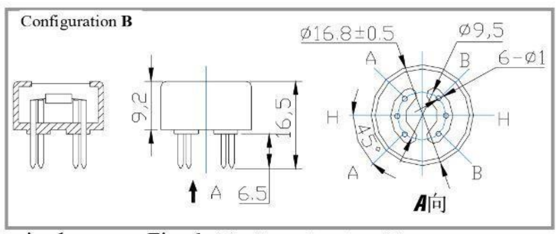


Fig.2





Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration **A or B**), sensor composed by micro AL2O3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

TEL: 86-371-67169070 67169080 FAX: 86-371-67169090 E-mail: sales@hwsensor.com

sensitive components. The enveloped MQ-135 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

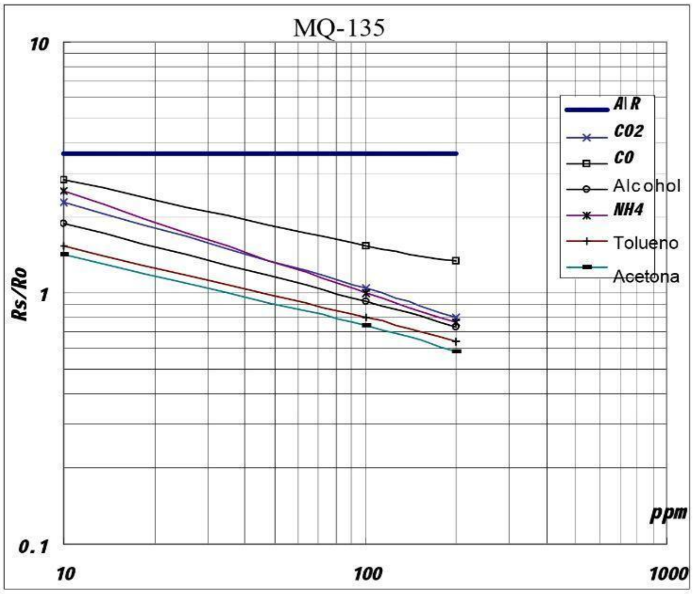


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases.
in their: Temp: 20℃、
Humidity: 65%、
O_2 concentration 21%
RL=20k Ω
Ro: sensor resistance at 100ppm of NH_3 in the clean air.
Rs:sensor resistance at various concentrations of gases.

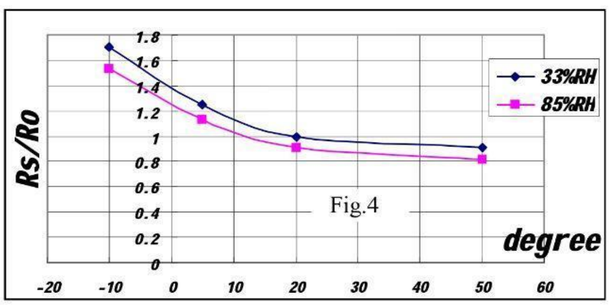


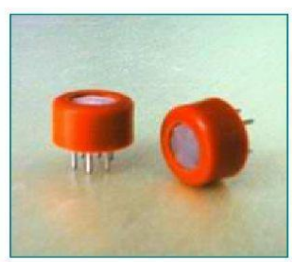
Fig.4

Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity.
Ro: sensor resistance at 100ppm of NH_3 in air at 33%RH and 20 degree.
Rs: sensor resistance at 100ppm of NH_3 at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So,When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 100ppm NH_3 or 50ppm Alcohol concentration in air and use value of Load resistancethat(R_L) about 20 K Ω (10K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.



Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixature, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2Sz, SO_X, Cl_2, HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

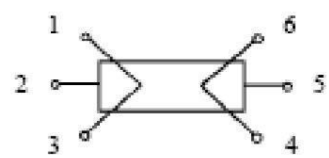
Do avoid icing on sensor'surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, if will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stbility before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

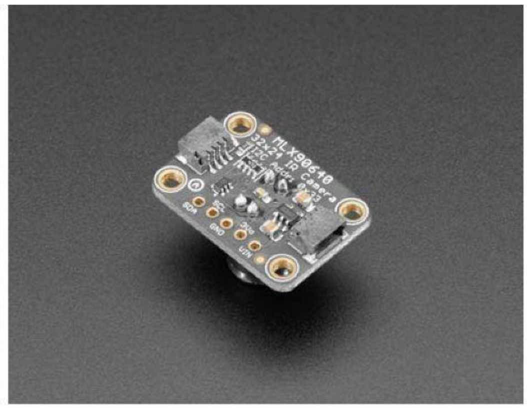
If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine
2.7.2 Speed: 1-2 Meter/ Minute
2.7.3 Warm-up temperature：100±20℃
2.7.4 Welding temperature：250±10℃
2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.



Adafruit MLX90640 IR Thermal Camera Breakout - 55 Degree

PRODUCT ID: 4407

You can now add affordable heat-vision to your project and with an Adafruit MLX90640 Thermal Camera Breakout. This sensor contains a 24x32 array of IR thermal sensors. When connected to your microcontroller (or Raspberry Pi) it will return an array of 768 individual infrared temperature readings over I2C. It's like those fancy thermal cameras, but compact and simple enough for easy integration.

This version has a narrow 55°x35° field of view we also have a version with a wider 110°x70° field of view

This part will measure temperatures ranging from -40°C to 300°C with an accuracy of +- 2°C (in the 0-100°C range). With a maximum frame rate of 16 Hz (the theoretical limit is 32Hz but we were not able to practically achieve it), It's perfect for creating your own human detector or mini thermal camera. We have code for using this sensor on an Arduino or compatible (the sensor communicates over I2C) or on a Raspberry Pi with Python. If using an Arduino-compatible, you'll need a processor with at least 20KB RAM - a SAMD21 (M0) or SAMD51 (M4) chipset will do nicely. On the Pi, you can even perform interpolation processing with help from the SciPy python library and get some pretty nice results!

This sensor reads the data twice per frame, in a checker-board pattern, so it's normal to see a checker-board dither effect when moving the sensor around - the effect isn't noticeable when things move slowly.

To make it easy to use, we hand-soldered it on a breakout board with a 3.3V regulator and level shifting. So you can use it with any 3V or 5V microcontroller or computer. We've even included SparkFun qwiic compatible STEMMA QT connectors for the I2C bus so you don't even need to solder! Just plug-n-play with any of our STEMMA QT (JST SH) cables.

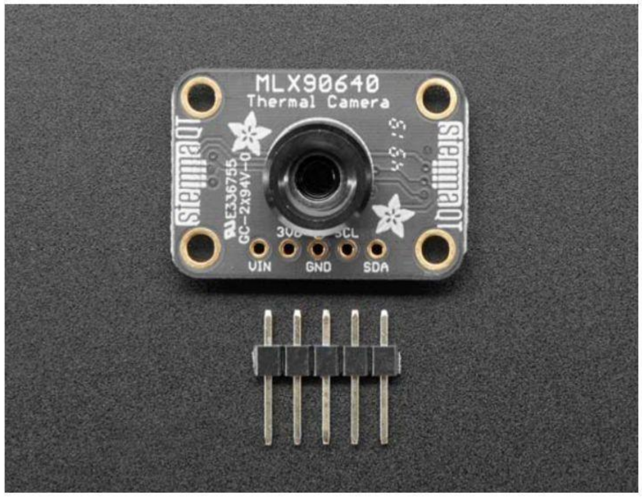
Even better - We've done all the hard work here, with example code and supporting software libraries to get you up in running in just a few lines of Arduino or Python code

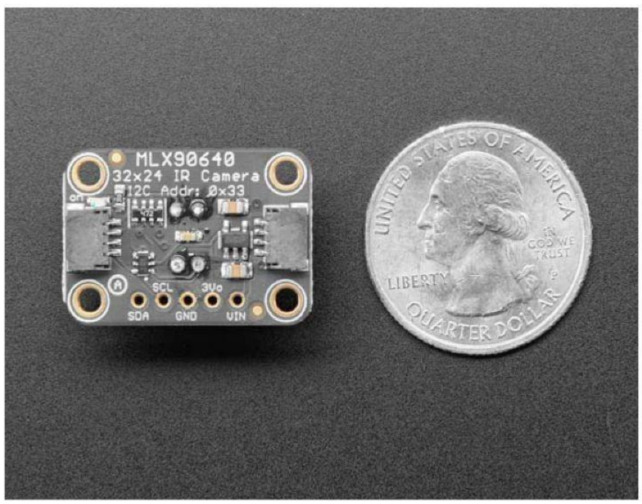
TECHNICAL DETAILS

- I2C compatible digital interface
- Programmable refresh rate 0.5Hz...64Hz (0.25 ~ 32 FPS)
- 3.3V-5V supply voltage, regulated to 3.3V on breakout
- Current consumption less than 23mA
- Field of view: 55°x35°
- Operating temperature -40°C ÷ 85°C
- Target temperature -40°C ÷ 300°C

Product Dimensions: 25.7mm x 17.7mm x 16.0mm / 1.0" x 0.7" x 0.6"

Product Weight: 3.5g / 0.1oz





https://www.adafruit.com/product/4407/1-6-20

ANNEX D

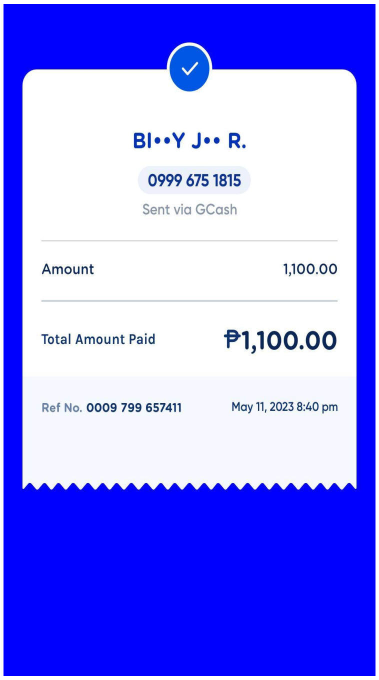
Bill of Materials and Retrieved Receipts

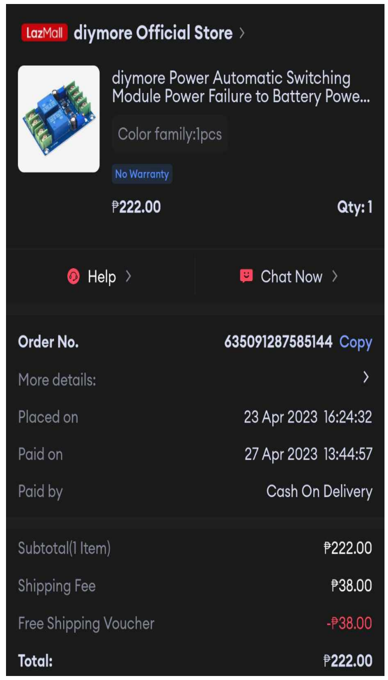
Table of Materials

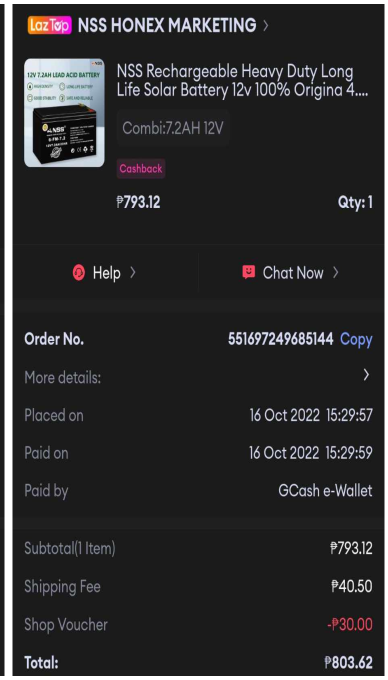
Materials	Quantit y	Amount (₱)
12-V Battery	1	800.00
MQ135 Air Quality Sensor	1	80.00
MLX 90640 Thermal Camera	1	3,500.00
Raspberry Pi 4, 8gb	1	8500.00
Microsoft Lifecam	1	1600.00
3D Printed Sensor Casing	1	700.00
LM2596s DC-DC Step-down Buck Converter Module	1	80.00
Aluminium Casing	1	500.00
PCB	1	50.00
Male & Female Wire Connectors	1 pack	60.00
TOTAL		**15,870.00**

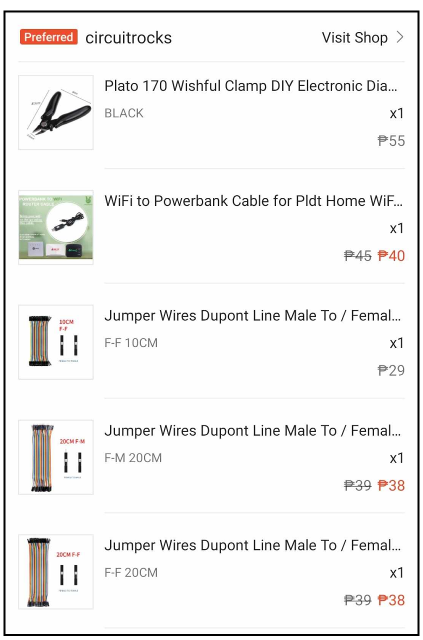
Retrieved Receipts

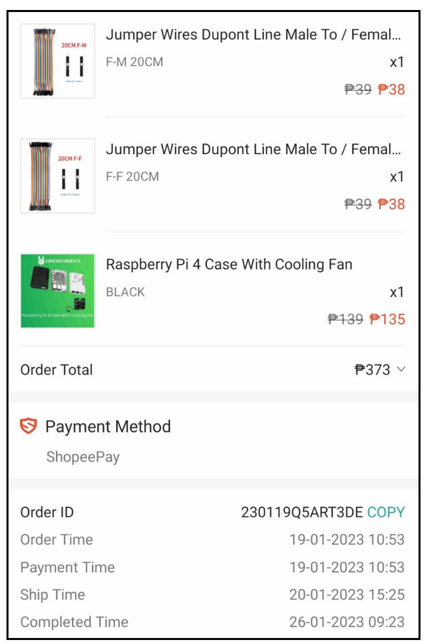
3D Printing Receipt


Bl••Y J•• R.
0999 675 1815
Sent via GCash
Amount 1,100.00
Total Amount Paid ₱1,100.00
Ref No. 0009 799 657411 May 11, 2023 8:40 pm


LazMall diymore Official Store
diymore Power Automatic Switching Module Power Failure to Battery Powe...
Color family:1pcs
No Warranty
₱222.00 Qty: 1
Help
Chat Now
Order No. 635091287585144 Copy
More details:
Placed on 23 Apr 2023 16:24:32
Paid on 27 Apr 2023 13:44:57
Paid by Cash On Delivery
Subtotal(1 Item) ₱222.00
Shipping Fee ₱38.00
Free Shipping Voucher -₱38.00
Total: ₱222.00

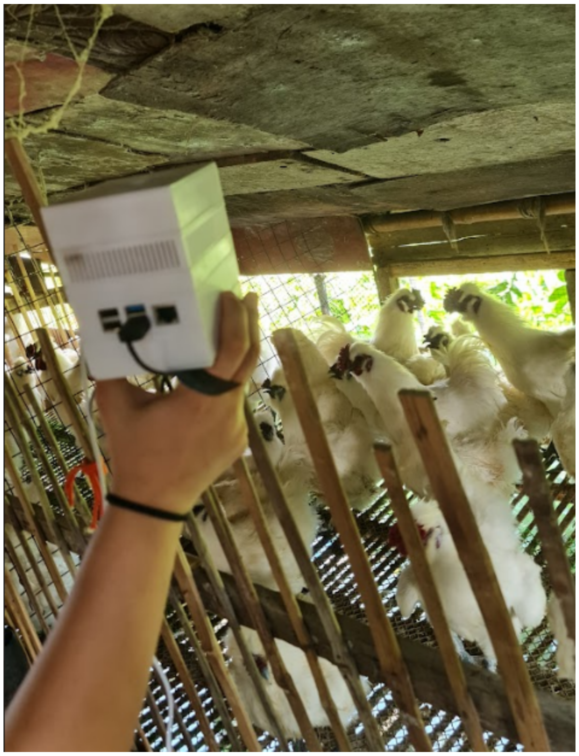
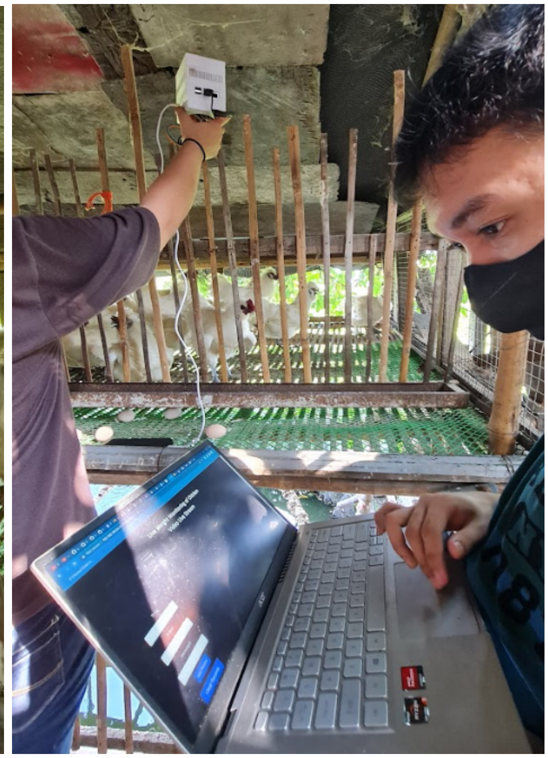

LazTop NSS HONEX MARKETING
NSS Rechargeable Heavy Duty Long Life Solar Battery 12v 100% Origina 4....
Combi:7.2AH 12V
Cashback
₱793.12 Qty: 1
Help
Chat Now
Order No. 551697249685144 Copy
More details:
Placed on 16 Oct 2022 15:29:57
Paid on 16 Oct 2022 15:29:59
Paid by GCash e-Wallet
Subtotal(1 Item) ₱793.12
Shipping Fee ₱40.50
Shop Voucher -₱30.00
Total: ₱803.62

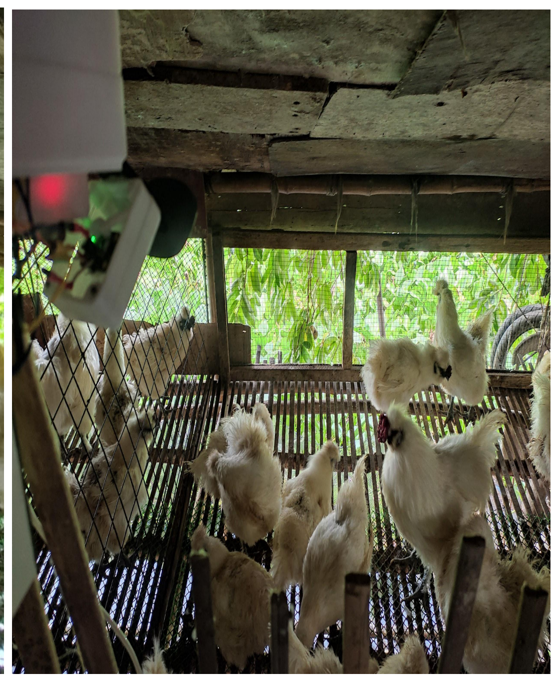

Preferred circuitrocks Visit Shop
Plato 170 Wishful Clamp DIY Electronic Dia...
BLACK x1
₱55
WiFi to Powerbank Cable for Pldt Home WiF...
x1
~~₱45~~ ₱40
Jumper Wires Dupont Line Male To / Femal...
F-F 10CM x1
₱29
Jumper Wires Dupont Line Male To / Femal...
F-M 20CM x1
~~₱39~~ ₱38
Jumper Wires Dupont Line Male To / Femal...
F-F 20CM x1
~~₱39~~ ₱38

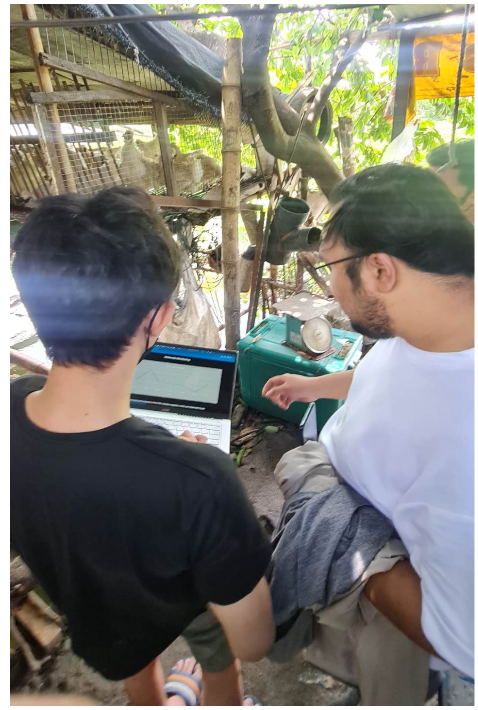

Jumper Wires Dupont Line Male To / Femal...
F-M 20CM x1
~~₱39~~ ₱38
Jumper Wires Dupont Line Male To / Femal...
F-F 20CM x1
~~₱39~~ ₱38
Raspberry Pi 4 Case With Cooling Fan
BLACK x1
~~₱139~~ ₱135
Order Total ₱373
Payment Method
ShopeePay
Order ID 230119Q5ART3DE COPY
Order Time 19-01-2023 10:53
Payment Time 19-01-2023 10:53
Ship Time 20-01-2023 15:25
Completed Time 26-01-2023 09:23

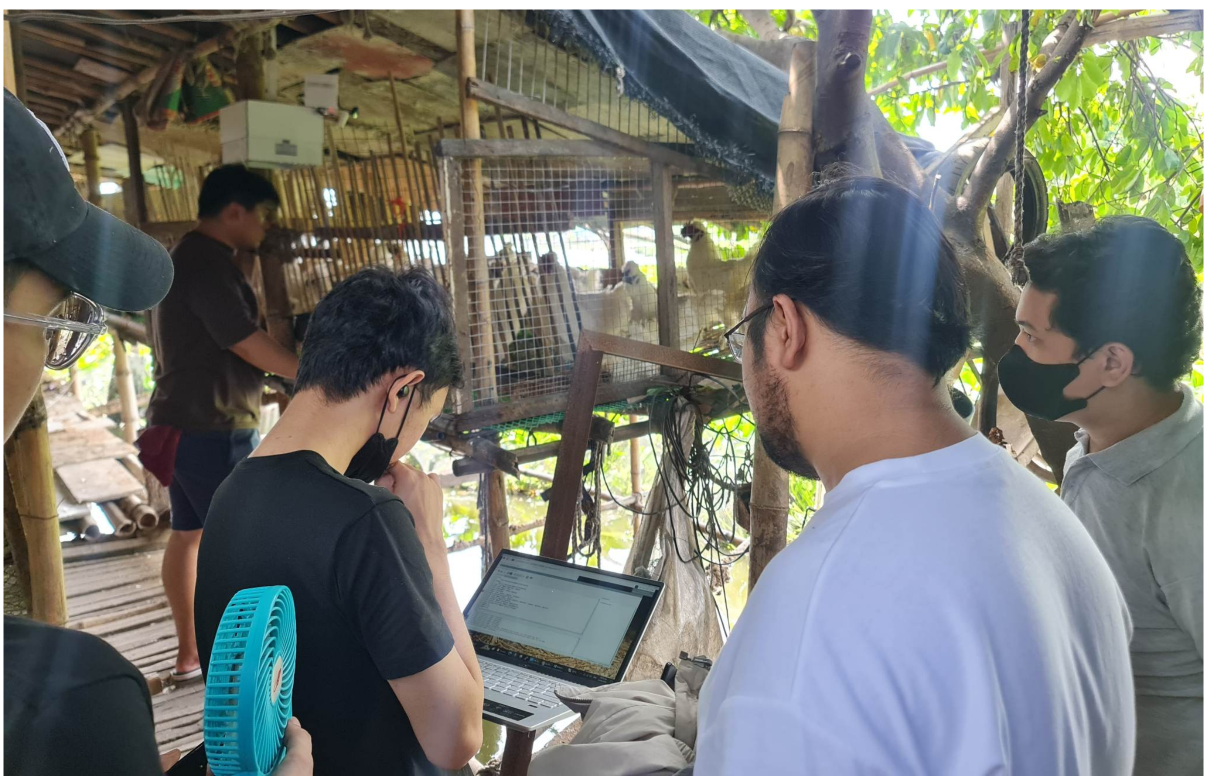
ANNEX E

Documentation








Asahi

Asahi



DONT BLAME
THE KIDS


ROBIE
C.M.

ANNEX F

User's Manual

GALLUSPY CAMERA

REAL-TIME MONITORING SYSTEM

OF BODY CHARACTERISTICS

AND BEHAVIOR OF *GALLUS DOMESTICUS*

USING COMPUTER VISION VIA

INTERNET-OF-THINGS (IOT)

USER'S MANUAL

TABLE OF CONTENTS

GALLUSPY CAMERA USER MANUAL

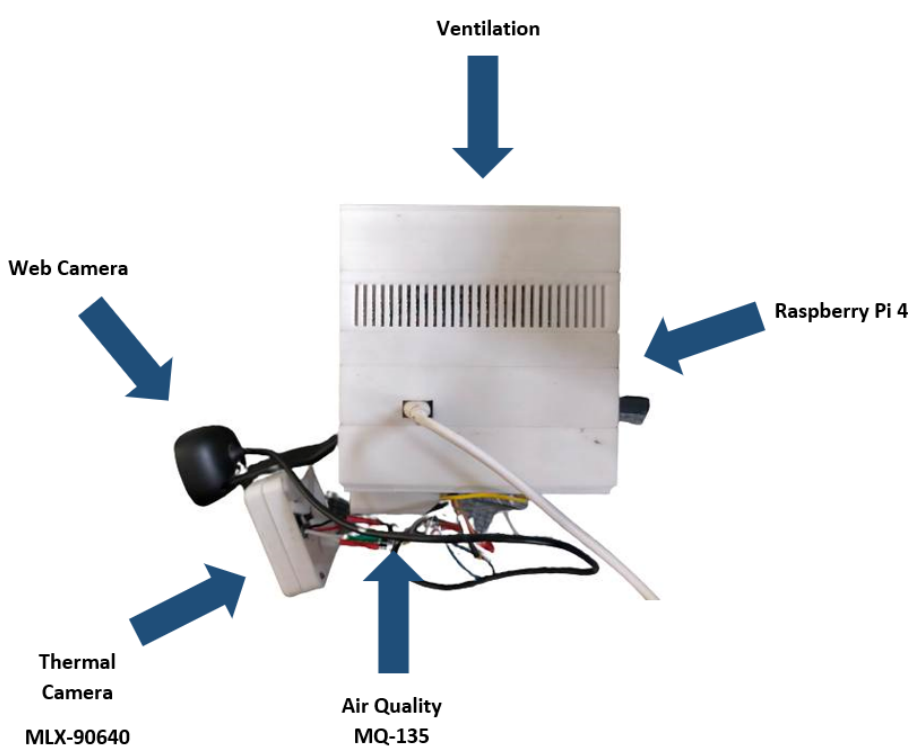
OVERVIEW

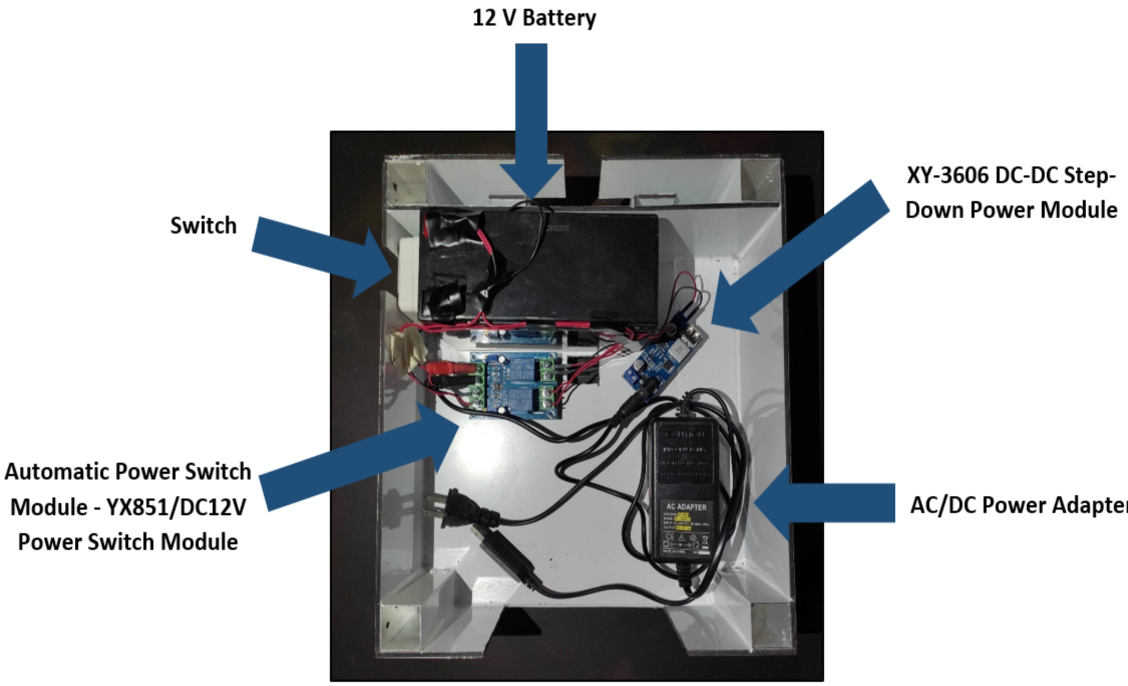
The GalluSpy Cam, a real-time monitoring system designed for *Gallus domesticus*. Combining computer vision technology and Internet-of-Things (IoT) connectivity, GalluSpy Cam allows poultry farmers, researchers, and enthusiasts to monitor the body characteristics and behavior of chickens with unparalleled precision.

The manual provides detailed instructions and step-by-step procedures for setting up, installing, and operating GalluSpy Cam. You will learn how to connect the cameras, configure the software, and harness the system's full potential. Discover how GalluSpy Cam analyzes body characteristics, detects behaviors, and provides real-time data and notifications. With GalluSpy Cam, you can explore the world of *Gallus Domesticus* like never before.

PARTS OF THE DEVICE

This section provides the parts and its description of the device, recognizing the functionality of each part shown.





1. **Web Camera -** The web camera captures and monitors the activities of the chickens. It captures the weight and mortality rate of the chickens.
2. **Thermal Camera (MLX-90640) -** The thermal camera captures the body temperature of the chickens and the surrounding temperature of the coop.
3. **Air Quality Sensor (MQ-135) -** The air quality sensor monitors the level of ammonia inside the coop.
4. **Raspberry Pi 4 -** It serves as the microcomputer that runs the system program.
5. **12 V Battery -** It is the back-up battery that will provide power to the device in case of power outage.
6. **Automatic Power Switch** - It enables switching between the 12-V Battery and the 220 V AC outlet.
7. **Switch -** It is located at the right side of the device that enables the user to turn on and off the device.

PLACEMENT OF THE DEVICE INSIDE THE COOP

1. Take into account the structure of your chicken coop and the locations you want to monitor. Find a location on the ceiling that provides a decent perspective point and covers the required areas.
2. Make sure the camera is correctly positioned so it can capture the activities inside the coop accurately.
3. To prevent it from dangling or getting damaged, make sure that you secure the electrical wire along the ceiling or wall using cable clips or tubes. Make that the camera is securely connected to the power supply.
4. Adjust the camera's angle to capture the desired field of view once it has been securely installed. To ensure that the camera covers the entire chicken coop, you may have to make adjustments with different perspectives.
5. Test the camera after installation to make sure it is operating properly. Verify the power supply is functioning properly, that the camera successfully captures the appropriate footage, and the video feed. Adjustments and troubleshooting should be made as necessary.

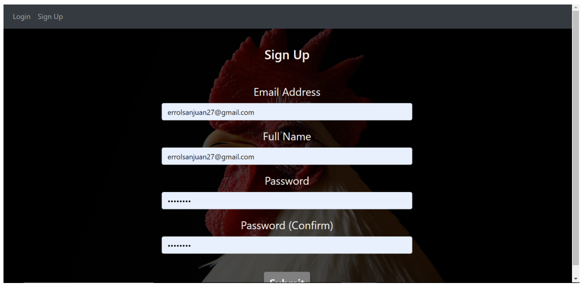
USING THE WEBSITE APPLICATION

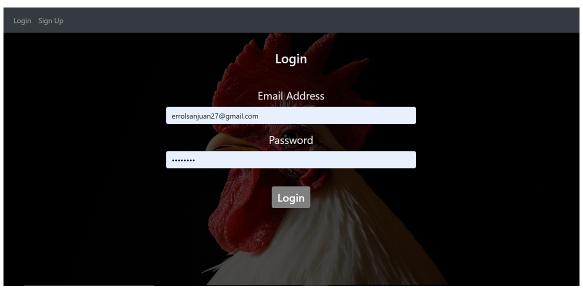
This section shows the instructions on how to use and navigate the website application. This section covers different aspects of the website application such as the login, sign-up form, about the system, data monitoring, preventions, downloadable history logs and contact information.

Creating an Account

Create an account, input your email,full name and password.

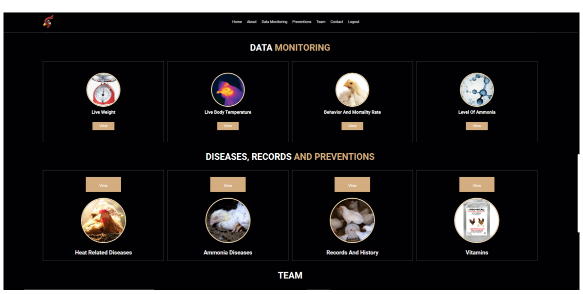
After the user fill-out the his/her personal email, full name and password, the user may now log in to his/her account.



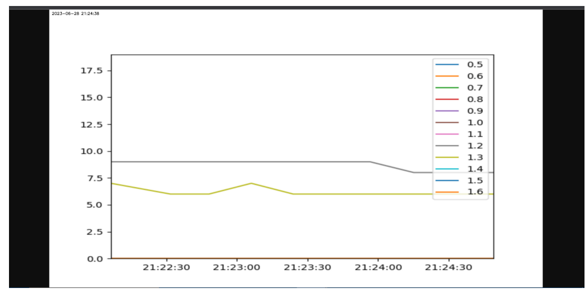


Navigating the Homepage

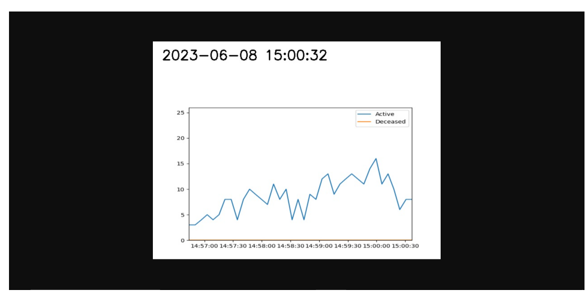
The user may now navigate the homepage of Galluspy. Each header is redirected to about the device, data monitoring, preventions, team, and contact.



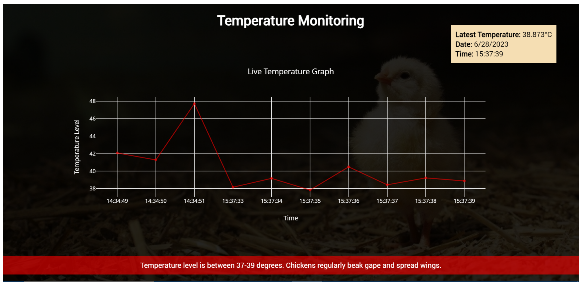
By clicking the "Live Weight" view button, the user can monitor the live weight through the camera and also observe the changes through the graph.



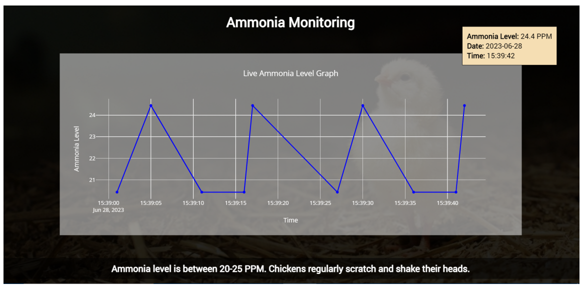
By clicking the "Behavior and Mortality Rate" view button, the user can monitor the live mortality rate of the chickens and also observe the changes through graphs.



By clicking the "Live Body Temperature" view button, the user can monitor the live body temperature of the chickens and also observe the changes through graphs and parameters.



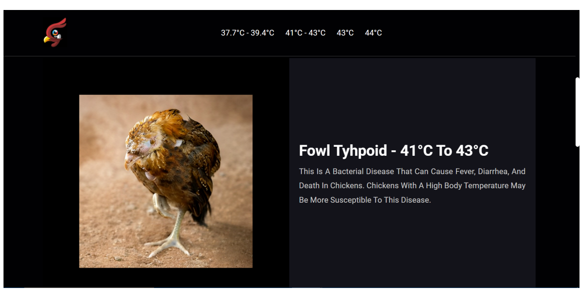
By clicking the "Level of Ammonia" view button, the user can monitor the current level of ammonia through parameters and graphs.



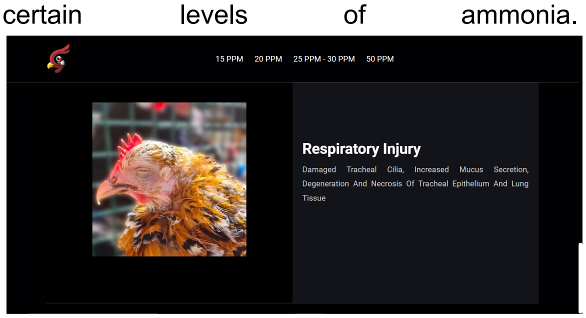
Other Features of the Website Application

(This is the Preventions Section)

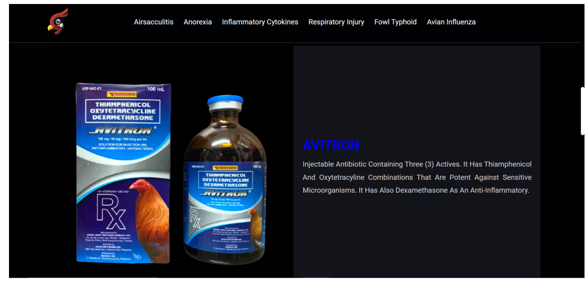
By clicking the "Heat Related Diseases" view button, the user can view the possible diseases at certain temperatures.



By clicking the "Ammonia Diseases" view button, the user can view the possible diseases at certain levels of ammonia.

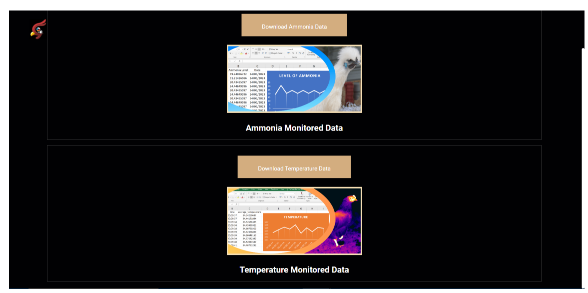


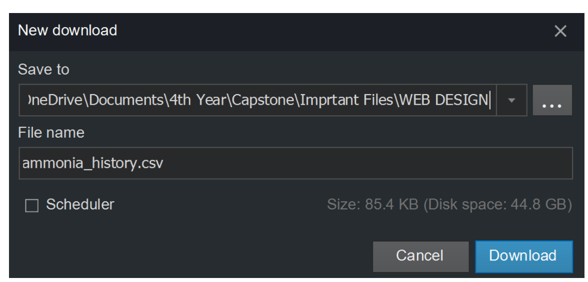
By clicking the "Vitamins" view button, the user can view the available vitamins and medicines for certain diseases related to the ammonia level and temperature.



Downloading the monitored data (CSV file)

By clicking the "Records and History" view button, the user can download the csv files with recorded ammonia level and temperature level including the time and date.





SAFETY PRECAUTIONS

This section shows the important safety precautions on using the GalluSpy Cam. Understanding these safety measures will ensure the device's durability and safety.

1. Do not expose the device to liquids, moisture, or water. Any component should not be exposed in water.
2. To avoid unintentional breakage or damage, handle the device and its parts carefully. Do not drop the device or hit it on a hard surface.
3. Use the device only for the purposes specified in the user manual. Without proper consent from the manufacturer, do not change or modify the device.

MONITORING AND MAINTENANCE

This section of the manual focuses on instructing users on how to efficiently maintain the functionality of the device.

1. Regularly check the device for physical damage, loose wire connections, and wear by visually looking for these indications. Inform the researchers of any problems.
2. Inspect the device's connection while it is functioning to make sure the chickens and other parameters are still being recorded and monitored.
3. Follow the troubleshooting instructions in the user manual to quickly address any issues. Take the necessary steps to fix the problem.
4. Use a clean, soft cloth and a light disinfectant to wipe out the device's surfaces.

TROUBLESHOOT AND SUPPORT

In this section, users can find help in recognizing and addressing any difficulties that might arise while operating their device. Valuable advice is provided regarding typical issues and suggested solutions.

Problem	Cause	Solution
The Device does not turn on. Power issues.	Power Supply Malfunction. Drained Battery or cord disconnected to the outlet.	1. Ensure the power cord is connected to the outlet. 2. Check if the battery is drained and needs charging or replacement. 3. Check for loose connections between the device and the power cord.
The device turns on but the website application does not read data.	The wifi is not connected.	1. Check the wifi connection. 2. Check if your device is connected to the wifi. 3. Contact the researchers for assistance in case the disconnection persists.
The sensors are working but do not collect data.	The sensors are not connected properly or sensor failure.	Contact the team for sensor replacement.
The device does not switch power sources during power outages.	Switching Module Failure	Contact the team for the replacement of the component.

TECHNICAL SPECIFICATIONS

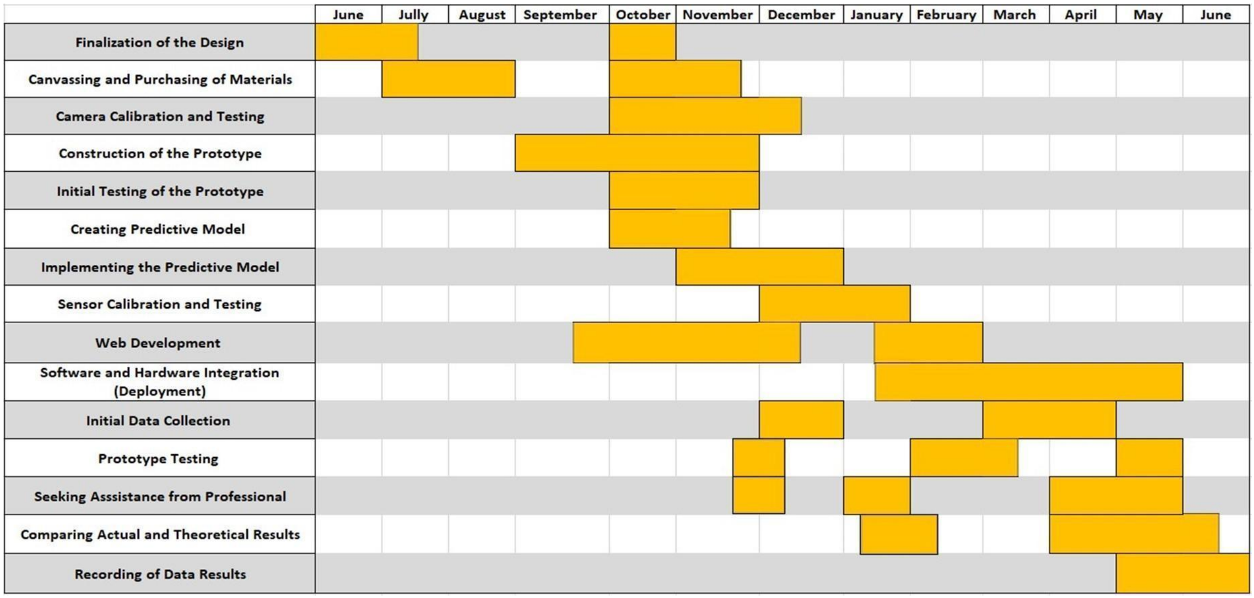
In this section, important technical information and device specifications are summarized briefly.

DEVICE SPECIFICATIONS	
Device Name:	GalluSpy Camera
Dimensions:	150MM x 90MM x 60MM
12-V Battery Charging Time:	6 to 8 Hours
12-V Battery Usage Time:	20 Hours
12-V Battery Capacity:	7.2AH Lead Acid Battery
Connection:	Wi-Fi
Air Quality Sensor Ammonia Range:	10 PPM to 300 PPM
Thermal Camera Resolution:	32x24 pixels
Thermal Camera Temperature Range:	-40ºC to 300ºC
Microsoft LifeCam Resolution:	1280 x 800 pixels

ANNEX G

Gantt Chart

Project Timeline



ANNEX H

Curriculum Vitae

JERICHO O. AVENIDO

Undergraduate Electronics Engineering

Enthusiastic and driven Electronics Engineering student with strong work ethic, problem-solving aptitude and excellent communications skills. Currently seeking an internship position which deals with passion for data communications, telecommunication networks design and software programming to leverage on honing further abilities based on corporate best practices and globally accepted technology standards.

Contact

+639602471788

agoncillo.avenido@gmail.com

@https://www.linkedin.com/in/jericho-avenido-a629a9145/

Personal Details

Date of Birth
28th June, 2000

Nationality
Filipino

Marital Status
Single

Address
Blk G Lot-26 Mayon St.,
ACM City of Imus, Cavite
4103

Education

TERTIARY
Bachelor of Science in Electronics Engineering,
Technological University of the Philippines - Manila
(2019 - 2023)

SECONDARY
STEM Strand, Technological University of the Philippines - Cavite
(2017-2019)
Imus Institute of Science and Technology
(2013-2017)

PRIMARY
Imus Pilot Elementary School
(2007-2013)

I hereby certify that the written information is true and correct to the best of my knowledge.

JERICHO O. AVENIDO
Applicant

Work Experiences

Assistant Project Coordinator at Domaintricks Corporation
July 2022-September 2022

- Monitoring the activities and accomplishments of Site Engineers (Wireless - SMART & GLOBE)
- Consolidate all updates of the project using ISDP for Quality Control Checking
- Tracking Site and Record Activity - ISDP HUAWEI & Perform Basic Functionality Testing (BFT)

Certifications

- Master IP Addressing and Subnetting for CCNA
- Neural Network and Natural Language Processing
- Amazon Web Service Cloud Practitioner
- Fortinet Network Security Expert Level 1,2, and 3
- $(ISC)^2$ Candidate

Skills

- Critical Thinking
- Teamwork and Collaboration
- Flexible and Adaptable
- Planning and Coordination
- Excellent Communication
- Dependable and Responsible

MARK ZIAN LOUIZE L. BUBAN

Undergraduate Electronics Engineering

An Electronics Engineering student seeking for an internship in relation to web development and software programming to earn new skills, expand my knowledge, and leverage my learnings. To look for an opportunity to be a part of an organization where I can use my problem solving skills and creativity to develop new technologies.

Contact

+639658417753
mrkznlze57899@gmail.com
https://www.linkedin.com/in/mark-zian-louize-buban-2b75a823b/

Work Experience

Work Immersion
Christ the King Medical Center Unihealth
130 Real St., Las Piñas City
January 2019

Seminars and Certifications

- Hack-Proof Your Life: A Beginner's Guide to Information Security - *June 2023*
- Unlocking the Limitless: Getting Cloud-Powered with Google Cloud Platform - *June 2023*
- APPRECIATE 2023: Annual Presentation of Project Research in Electrome-chanical, Civil, Information and Telecommunications Engineering - *May 2023*
- App Dev Insider: Behind the Scenes of Application Development using Flutter - *May 2023*
- Training for Master IP Addressing and Subnetting for CCNA - *May 2022*

Skills

- Basic Electronic Skills
- Programming Languages Basics
- Proficient Computer Literacy
- Written and Verbal Communication
- Mathematical Reasoning
- Adaptability and Flexibility

Personal Details

Date of Birth
20th June, 2001

Nationality
Filipino

Marital Status
Single

Address
Blk. 20 Lot 18, Bernabe Compound, Pulanlupa I, Las Piñas City

Education

TERTIARY
Bachelor of Science in Electronics Engineering
Technological University of the Philippines - Manila
(2019 - present)

SECONDARY
Science, Technology, Engineering. and Mathematics Strand, Las Piñas National High School - SHS Main Campus
(2017-2019)
Las Piñas National High School - MAIN
(2013-2017)

PRIMARY
Pulanlupa Elementary School
(2007-2013)

I hereby certify that the written information is true and correct to the best of my knowledge.

MARK ZIAN LOUIZE L. BUBAN
Applicant

NICOLE ANNE C. CATUDIO

Undergraduate Electronics Engineering

A team-player and self-motivated graduating student looking for a chance to apply to a program in line with Telecommunication, Cyber Security, and other fields, which I believe can give me the opportunity to strengthen my knowledge, abilities, and skills needed in a real-world job. Hard work, flexibility, determination, and the will to learn and adapt to new things are some of the most important traits that I believe I possess in order to survive in this continuously changing environment.

Contact

+639426034207
catudion@gmail.com
linkedin.com/in/nicole-anne-catudio

Personal Details

Date of Birth
25th June, 2000

Nationality
Filipino

Marital Status
Single

Address
37 Damzon St. Dampalit
Malabon City

Education

TERTIARY
Bachelor of Science in Electronics Engineering,
Technological University of the Philippines - Manila
(2019 - 2023)

SECONDARY
STEM Strand, University of the East - Caloocan
(2017-2019)
Malabon National High School
(2013-2017)

PRIMARY
Academia de Santo Rosario
(2007-2013)

Work Experiences

IT Intern and Technical Support at WSAP Inc.

August 2022 - September 2022

- Attended morning briefings and accomplished daily tasks and reports.
- Facilitates and monitor the companies meetings and event through out the day.
- Provide assistance and troubleshooting for technical issues and inquiries related to the company's meetings and events

Certifications

- Master IP Addressing and Subnetting for CCNA
- Exploring the Role of Open RAN in Today's Communication
- SAP Analytics Cloud Workshop
- Fortinet Network Security Expert Level 1,2, and 3
- (ISC)² Candidate
- Hack-Proof Your Life: A Beginner's Guide to Information Security
- App Dev Insider: Behind the Scenes of Application Development using Flutter
- Diving into Depths of Understanding Circuits

Skills

- Critical Thinking
- Organization & Planning
- Time Management
- Microsoft Office
- Cisco Packet Tracer
- Matlab & Octave
- Multisim & Proteus
- MS Dos & Basic Python

I hereby certify that the written information is true and correct to the best of my knowledge.

NICOLE ANNE C. CATUDIO
Applicant

JOHN VHINCENT RUIZ

Undergraduate Electronics Engineering

I am a student of Bachelor of Science in Electronics Engineering with a hardworking, productive, and flexible personality, seeking trainings under Electronics, IT, Telecommunication and Technology Industry that would help to improve skills and grow in the chosen field.

Contact

- +639271556354
- jvhincentruiz@gmail.com
- www.linkedin.com/in/john-vhincent-ruiz-6b730b244

Personal Details

Date of Birth
8th February, 2001

Nationality
Filipino

Marital Status
Single

Address
Phase2, Bllk2, Lot3, RCD Royale Homes, Kalubkob, Silang, Cavite

Work Experiences

Melham Construction Corporation
August 2022 - September 2022

- Technology proposal for the company for business purposes
- Attend company organized seminars
- Create value proposition for the proposed technology

Education

TERTIARY
Bachelor of Science in Electronics Engineering,
Technological University of the Philippines - Manila
(2019 - 2023)

SECONDARY
Infant Jesus Academy of Silang inc.
(2013-2019)

PRIMARY
Josephine F. Khonghun Special Education Center
(2007-2013)

Certifications

- Master IP Addressing and Subnetting for CCNA
- Fortinet Network Security Expert Level 1, 2 and 3
- (ISC)^{2}Candidate
- Hack-Proof Your Life: A Beginner's Guide to Information Security
- Exploring the role of Open RAN in Today's Communication

Skills

- Computer Literacy
- Good Communication
- Teamwork and Collaboration
- Basic Python Programming
- Willingness to Learn
- critical Thinking

I hereby certify that the written information is true and correct to the best of my knowledge.

JOHN VHINCENT RUIZ
Applicant

ERROL JAMES B. SANJUAN

Undergraduate Electronics Engineering

An Electronic Engineering student who is passionate about data communications, telecommunications network design, and software programming to be used to hone additional abilities based on the company practices. Dedicated and driven work ethic with excellent communication, coordination, time management, dependability, and critical thinking abilities.

Contact

09060219951
errolsanjuan07@gmail.com
https://www.linkedin.com/in/errol-james-san-juan-824587244/

Personal Details

Date of Birth
22nd July, 2000

Nationality
Filipino

Marital Status
Single

Address
1164 Labores St. Brgy. 848, Pandacan, Manila

Education

TERTIARY
Bachelor of Science in Electronics Engineering
Technological University of the Philippines - Manila
(2019 - 2023)

SECONDARY
Marinduque Midwest College
(2013-2017)

PRIMARY
Masiga Elementary School
(2007-2013)

Work Experiences

Assistant Project Coordinator at Domaintricks Corporation
August 2022-November 2022

- Monitoring the activities and accomplishments of Site Engineers (Wireless - SMART & GLOBE)
- Consolidate all updates of the project using ISDP for Quality Control Checking
- Tracking Site and Record Activity - ISDP HUAWEI & Perform Basic Functionality Testing (BFT)

Certifications

- Master IP Addressing and Subnetting for CCNA
- Neural Network and Natural Language Processing
- Introduction to Packet Tracer
- Exploring the role of Open RAN in Today's Communication
- Diving into Depths of Understanding Circuits

Skills

- Electronic Circuit Analysis and Design
- Structural Modeling
- Software Programming
- Teamwork and Collaboration
- Planning and Coordination
- Dependable and Responsible

I hereby certify that the written information is true and correct to the best of my knowledge.

ERROL JAMES B. SAN JUAN
Applicant