TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Blvd., Ermita, Manila

COLLEGE OF ENGINEERING

ELECTRONICS ENGINEERING DEPARTMENT

electronics@tup.edu.ph

SWAMP: Flood Monitoring, Prediction, and Alert System Using Wireless Sensor Network and Internet of Things (IoT)

by

Bandiola, Carla Gaelle D.

Bognot, Lanz Lowen Jay O.

Calangian, Marwin R.

De Guzman, Edcelle Shane D.

Manalo, Jonathan Michael A.

Ventura, Harvey C.

August 2023

APPROVAL SHEET

This project entitled "**SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IOT)**", has been approved and submitted by the following proponents:

Bandiola, Carla Gaelle D.
Bognot, Lanz Lowen Jay O.
Calangian, Marwin R.
De Guzman, Edcelle Shane D.
Manalo, Jonathan Michael A.
Ventura, Harvey C.

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

Engr. Mark P. Melegrito
Project Adviser

Engr. Jessica S. Velasco
Panel Chair

Engr. Jomer V. Catipon
Panel Member

Engr. August C. Thio-Ac
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 Philippines is vulnerable to tropical cyclones due to its location, resulting in heavy rainfall, flooding areas, and high winds that cause severe damage to crop and property. This study introduces a scientific and efficient response to flooding. The System for Waterline Alert and Monitoring Probe (SWAMP) monitors, predicts, and alerts the community of the current flood level. This device uses wireless sensor networks to measure specific parameters: water level, rainfall, and dam information that causes floods and operates through the Wireless Sensor Networks (WSN) and Internet of Things (IoT). The system uses the YOLOv5 model, integrated with a web digital camera, to monitor the real-time water level of the area. The project incorporates a user-friendly website with notifications as part of the alert aspect and data display. The model has a training and validation accuracy of 97% with the gathered data from its camera. The Long Short-Term Memory (LSTM) cascaded with the RandomForest model used for prediction has a training and validation accuracy of 98%. The accuracy of the LSTM model is further validated by using Mean Absolute Percentage Error, yielding 1.18%, which indicates very low error in predicting flood levels.

ACKNOWLEDGEMENT

The proponents are grateful to their hardworking adviser, **Engr. Mark P. Melegrito**, for providing guidance and support for this study. Also, to **Engr. Nilo M. Arago**, our Project Study Professor and College Dean, for extending his knowledge and suggestions to develop our project study.

The researchers would also like to thank the **ECE Faculty**, especially the panel members who have given their sincere recommendations, and guidance that helped the proponents with the success of their study.

The proponents would also like to commend the **officials of Barangay Frances in Calumpit, Bulacan**, especially **Hon. Cecilia N. Agayan**, **Hon. Jessie C. Cayago**, **Mr. Warly LLave**, and the whole members of the **Sangguniang Barangay of Frances**. Furthermore, extending our appreciation to the **family of Hon. Jessie C. Cayago** for accommodating the researchers and granting their place for the project's deployment. As well as the families of the two additionally deployed nodes in the barangay.

The researchers also regard the **Philippine Atmospheric, Geophysical, and Astronomical Services Administration (PAGASA)** for lending the hydrological data needed for the project's algorithm. Also, acknowledge the **Pampanga River Basin Flood Forecasting and Warning Center (PRFFWC-PAGASA)** and **Mr. Hilton Hernando**, the chief of (PRFFWC-PAGASA), for providing the needed data for the project study.

Lastly, the proponents are very grateful to the people who have provided their help, knowledge, and guidance to finish this study.

- The Proponents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SUMMARY TABLES

CHAPTER 1

THE PROBLEM AND ITS BACKGROUND

1.1 Introduction

The Philippines barely exceeds the equator, so hurricane energy is absorbed to a minimum before reaching the shore. Due to the warm tropical sea, about 20 typhoons occur every year (FutureLearn, 2021).

The Philippines is a typhoon-prone area generally attributed to its geographic location. These typhoons commonly cause strong winds and heavy rainfalls that affect the daily living conditions of the people. It affects the production of crops, limits the farmers' harvests, causes landslides, and may cause severe damage to properties. Therefore, having enough knowledge about these occurrences is crucial to be prepared for the upcoming disaster and limit its potential damage. The southwest monsoon or hanging Habagat can be formed into a typhoon, commonly affecting the country from June to September. It is most active in August, while least active in May. The Philippine area of responsibility is located around vast oceans such as the South China Sea and the Pacific Ocean, which heavily influences the country's yearly occurrence of around 20 tropical cyclones. The Philippines is "the most susceptible country in the world to tropical storms," according to a 2013 Time Magazine story.

Bulacan is one of the most flood-prone states in the Philippines. Excessive groundwater extraction causes land subsidence. Bulakeños is at risk of severe flooding due to rising sea levels and subsidence due to global climate change (Bello, 2021). In addition, Bulacan is prone to flooding due to high tides and the discharge of water from dams around Calumpit, namely Angat and Pantabangan. Aside from the mentioned reasons of flooding, one of the barangays in Calumpit is suffering flood because of the rapidly increasing water level waterline of nearby rivers due to high tide. This causes some areas to experience flood daily. The researchers conducted a survey and Community Needs Assessment with the help of local officials.

After leading a Community Needs Assessment, the specialists effectively consulted 30 respondents of Barangay Frances, Calumpit Bulacan. 100 percent asserted that they encountered flooding because of different reasons. 94.44% of them encountered flood because their homes are situated in swamp regions; 94.44% expressed that it is because of weighty downpours and storms; 66.67% said that it is brought about by the water delivered by adjacent dams and 66.67% guaranteed that it is because of elevated tide. During hurricanes, the rapidly rising water levels of neighboring streams cause streak floods. With respect to the data of downpour expectation, 72.22% of the respondents rely upon the declaration of barangay authorities; 33.3% rely upon the degree of water in adjacent regions; 38.89% rely upon weather conditions gauge and 16.67% rely upon schedule technique.

This study was conceptualized based on "*Smart Flood Detection, Warning, and Surveillance Systems Using Image Processing*." While assessing community needs in Barangay Frances, researchers recognized the importance of flood monitoring, forecasting, and warning systems in the area. In response to the problems faced by Barangay Frances Calumpit, Bulacan, the researchers propose a flood monitoring, forecasting, and warning systems using wireless sensor networks and the Internet of Things (IoT).

1.2 Background of the Study

Calumpit, Bulacan is one of the places in the Philippines that is considered to have a lower place and a catch basin of the nearby rivers and dams in Bulacan, Pampanga, and Nueva Ecija. In addition, high tide, rain, and storms are just few of the many causes of flooding in the area. As a result, the people had various ways of detecting floods, including a "flood calendar" and basing on the announcements of the barangay officials. The integration of floods in the community has made them resilient, causing the lack of a proper evacuation system in their place, specifically in Barangay Frances. The information gathered by the researchers proposed a flood monitoring, alert, and prediction system for the barangay in line with a previous study, "*Smart Flood Detection, Alarm, and Monitoring System Using Image Processing*." The proposed research aims to develop a better and more accurate system for the community to have a better evacuation system. In preparation, the proponents gathered related studies to develop a system fit for the community to use and benefit from.

The study "*An Intelligent Flood Monitoring System for Bangladesh Using Wireless Sensor Network*" proposed a neuro-fuzzy based flood alert system using Wireless Sensor Network or WSN. Distributed sensor nodes use a low wireless personal area network to collect water level data, precipitation data, wind speed data, and barometric pressure data from selected locations. Sensor information is sent to the Distributed Alert Center via the Arduino microcontroller and XBee transceiver. The Distributed Alarm Center uses an XBee transceiver and a Raspberry Pi microcomputer to generate flood alarms based on sensor information. The proponents used 20 years of flood monitoring data to estimate the duration of the flood and stored this data in a database.

The Raspberry Pi microcomputer creates the Intelligent NFC that uses sensor data to send flood alerts (Sakib et al., 2016).

The purpose of the study Wireless Sensor Nodes for Flood Forecasting using Artificial Neural Network aims primarily to predict water levels in high-risk areas of Masantol, Pampanga, using artificial neural networks. It intends to collect data about the target area using a wireless sensor node (WSN). It uses a microcontroller and GSM module for data acquisition and determine a neural network model for flood level prediction. The position of WSNs is in medium and excessive danger regions of Masantol, Pampanga. The proponents used an ultrasonic sensor for every sensor station to decide the actual time water stage. Every sensor dispatches data thru the GSM module to a few recipients: the administrator, nearby authorities' unit, and the DRRMO. To expect the water stage on every sensor station, the Time Series Neural Network was used. The Nonlinear autoregressive (NAR) and Nonlinear autoregressive with Exogenous input (NARX) models executed the modeling and prediction (Sahagun et al., 2017).

The study "*Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks*" designed and implemented an IoT-based flood monitoring system, calculating the alert system and the time for flooding to occur, and setting aside time for people to evacuate accordingly. It predicts rainfall levels using a machine learning model. It consists of various hardware components such as a Wi-Fi module, TFT display, water sensor, and raindrop sensor. Connecting all these components Raspberry Pi on PCB, three raindrop Sensors and water level sensors have been placed. Water level, water area, a buzzer, a warning message, and the remaining time of the flood zone are displayed. The flood monitoring and warning system is built by connecting IoT hardware to the Gecko platform and allows it to be monitored from a remote location. Linear Regression, Support Vector Machine and Artificial Neural Networks are the three different types of programs to predict floods for the next consecutive month. Different visualizations of data helped in implementing the approaches for the right prediction. The observation indicates machine learning models won't work well for the prediction of rainfall due to fluctuations in rainfall (Simatupang et al., 2019).

Another study named "*Design and Development of Flood Monitoring and Early Warning System*" comprises a hydreon optical rain sensor, GSM modem, temperature humidity sensor, and Arduino Uno. The project's primary goal is to produce effective monitoring of weather and alert people in the afflicted region. The raindrop sensor and temperature humidity sensor detect at a specific temperature, and the humidity level initiates the system and causes the Arduino microcontroller to send out a message GSM modem to message the selected location. The final output sends the message to the registered user, and the sensor's input matches the condition. The system conducted multiple tests to ensure that the information was correct and gathered sufficient data available to the registered user. The study recommends adding more input sensors in the future to collect more precise readings and realistic prediction outcomes depending on the water level in the chosen area (Dondang et al., 2021).

The referenced study "*Real Time Flood Detection, Alarm and Monitoring System Using Image Processing and Multiple Linear Regression"* concentrates on detecting and monitoring the flood level. It considers three parameters: precipitation rate, flood level, and flow rate. This study used three sensors: Float switch, Rain Gauge, and Flow rate meter sensor. These sensors are responsible for detecting the parameters needed and connecting to the Arduino UNO. The analog signal sent by the Arduino UNO sends it to the Raspberry Pi, and the camera captures the image to detect the flood level. These data are transmitted to the cloud via the internet, to the database, and then to the Android mobile application. The system uses multiple linear regression to predict the flood based on its three parameters (Tolentino et al., 2019).

1.3 Research Gap

A flood monitoring, prediction, and alert system device is a combination of sensors and algorithm capable of determining the expected flood height, rate, and expected time of arrival at a certain point in time before the event happens. This monitoring, prediction, and alert system is an application of wireless sensor network (WSN). The implementation of such system requires complex programming and site visitations which will necessitate supplemental studies and time. Due to Calumpit, Bulacan's geographic nature, it is well established that the entire area is a catch basin which makes it susceptible to floods. Thus, developing a wireless sensor network that can monitor and predict flood levels and alert the residents of Calumpit, Bulacan is hereby proposed. However, calibration of sensors and data gathering will be the real challenge ahead of this project.

1.4 Research Objectives

This study aims to develop a flood monitoring, prediction, and alert system using wireless sensor network and internet of things for flood prone residences in Barangay Frances in Calumpit, Bulacan. Specifically, it aims:

1. To design and fabricate flood monitoring & predictive device with integrated hydrological sensors using Wireless Sensor Network (WSN) & Internet of Things (IoT)
2. To develop image processing using a web digital camera for water level monitoring system and develop a predictive model for water level using time series and dam information
3. To implement a broadcasting device using an integrated siren and a user-friendly website that displays data from the integrated sensors and dam information for flood monitoring and prediction with web-based notification as an alert system.
4. To test and evaluate the functionality, reliability, usability, efficiency, maintainability, and portability of the flood monitoring, prediction, and alert systems.

1.5 Significance of the Study

Flooding is affecting lives by causing loss of agricultural and economic opportunities, damage to properties, psychosocial effects, migration, health concerns, and hindrances to economic growth and development. Thus, it is one of the major concerns of the residents of Barangay Frances, Calumpit, Bulacan.

This study will provide residents with flood monitoring, prediction, and alert system using wireless sensor network and Internet of Things (IoT) which will allow them to become aware and notified regarding the current and future state of the rivers surrounding the area covered by the nodes. With a proper monitoring and alert system, they will be able to have sufficient time to prepare and manage their daily activities depending on the rate of water level.

SWAMP aims to enhance the existing knowledge and technology concerning flood monitoring prediction and alert systems. New possibilities and re-developments in flood monitoring, prediction, and alert systems may be developed with the help of SWAMP for future studies.

This study is under Disaster Risk Reduction and Climate Change Adaptation (DRR & CCA), Section V, of the Harmonized National Research and Development Agenda (2017-2022) of the Department of Science and Technology (DOST). Given that the study aims to give flood monitoring and alerting assistance to the residents of Barangay Frances, Calumpit, Bulacan.

It contributes to the Goal 11, Sustainable Cities and Communities: Make cities and human settlements inclusive, safe, resilient, and sustainable, of the Sustainable Development Goals of the United Nations.

1.6 Scope and Limitations

SWAMP is a system composed of three main features: monitoring, prediction, and alert system deployed at Barangay Frances, Calumpit, Bulacan. Through Wireless Sensor Network (WSN), the proposed system seeks to measure flood monitoring parameters such as flood level, amount of rainfall, and atmospheric pressure A Rain Gauge (WH-SP-RG), Ultrasonic Sensor (JSN-SR04T), Barometric Pressure Sensor (BMP-180), and a web digital camera make up the Wireless Sensor Network. The system will employ Raspberry Pi and NodeMCU to communicate with wireless sensor networks and collect the necessary data for monitoring and forecasting. A solar panel powers the device to offer a sustainable power supply, with a backup lead acid battery for redundancy. Image processing with Convolutional Neural Network architecture is utilized for flood prediction. Within 1-3 hours in advance, a Time-Series algorithm will be utilized for flood prediction. The system implements a hardware and software notification. The hardware notification is mainly employed using a siren. A website developed for the alert system provides sensor data, water updates from the Angat and Pantabangan Dams, evacuation maps and routes, flood warning levels, flood management suggestions, and emergency contact numbers. This study does not cover the prediction of flood duration in the impacted area. Due to budget constraints, the flood monitoring, prediction, and alert system will be confined to Barangay Frances.

1.7 Definition of Terms

1. **Barometric Pressure Sensor**

   The BMP180 barometric pressure sensor is a sensor that can be used to detect altitude, measure vertical velocity, temperature, atmospheric pressure, and even temperature.

2. **Cloud Database**

   A cloud database can be accessed over the Internet, and it offers various advantages since it is scalable and accessible.

3. **Convolutional Neural Network**

   A convolutional neural network (CNN) is a deep learning algorithm that is particularly effective for processing and analyzing visual data.

   a. **VGG16**

      Visual Geometry Group is a Convolutional Neural Network architecture with 16 convolutional layers centered on object detection and classification algorithms.

   b. **ResNet-50**

      Residual Network 50 is a 50-layer-deep Convolutional Neural Network that can train extremely deep neural networks with 150+ layers.

   c. **Inceptionv3**

      Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification.

d. EfficientNet

EfficientNet is a convolutional neural network architecture and scaling method that utilizes a compound coefficient to uniformly scale all depth/width/resolution dimensions.

4. Time Series: Autoregressive

A time series autoregressive (AR) model is a statistical model that describes the relationship between an observed time series and its lagged values. It is a widely used approach for analyzing and forecasting time series data.

CHAPTER 2

REVIEW OF RELATED LITERATURE AND STUDIES

The review of related literature plays a pivotal role in research studies. It involves exploring and synthesizing articles and materials authored by scholars and experts in their field. This meticulous examination establishes a foundation, expands understanding, and nurtures a commitment to lifelong learning within their chosen discipline.

2.1 Convolutional Neural Network for Flood Monitoring & Prediction

2.1.1 Convolutional Neural Network for Flood Depth Approximation

The research study titled "*Detection of Flood Disaster System Based on IoT, big Data and Convolutional Deep Neural Network"* (Song et al., 2021) focuses on the management of disasters by leveraging IoT technologies to monitor, analyze, manage, and detect environmental conditions through real-time data. In this study, a Convolutional Deep Neural Network (CDNN) is employed for flood detection. The system incorporates flood-related big data, including water flow, water level, humidity, rain sensor readings, and other variables, which are processed using Hadoop Distributed File System and Map Reduce. Data cleaning techniques are applied to remove duplicate entries, followed by missing value imputation and normalization functions for preprocessing the collected data. The proposed CDNN approach yields various performance metrics such as Sensitivity, Specificity, Accuracy, Precision, Recall, and F-score.

The research study titled "*Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions*" (Kimura et al., 2019) aimed to investigate the application of convolutional neural networks (CNN) in flood prediction models. In this study, time series datasets were transformed into image datasets through a process that involved identifying the minimum and maximum values of water level and rainfall level. These values were then converted into a black-and-white image with gradation ranging from zero to one. The researchers selected two specific locations for testing purposes: the first location was characterized by heavy rainfall events caused by typhoons, while the second location experienced fewer instances of rainfall. The acquired rainfall and water level datasets spanned the period from 1992 to 2019. To enhance prediction efficiency, a transfer-learning approach was employed. The study focused on predicting the three most significant flood events and a moderately high flood event. Remarkably, the study successfully predicted time-series floods with a lead time of one hour. The findings demonstrated that CNN proved to be an effective algorithm for highly accurate predictions, and the errors decreased as the transfer-learning approach underwent more re-training iterations.

The research study titled "*Automatic flood detection in SentineI-2 images using deep convolutional neural networks*" (Jain et al., 2020) introduces an autonomous flood detection technique that combines a modified water index approach with a deep convolutional neural network (CNN) model. The technique is developed and evaluated using labeled high-resolution photos obtained from Sentinel-2 satellites. Prior to training and evaluation, a series of water index maps are generated for each dataset by processing the spectral images. These maps consist of 512 × 512 arrays, wherein each element represents the water index value for one of the six index types. To improve the accuracy of shallow water detection by mitigating the impact of clouds and cloud shadows, the modified water index method incorporates two indices: green/SWIR and blue/NIR. Among the models considered, the VGG16 CNN model yields the best performance when training the water index. The remarkable results obtained in this study highlight the potential of automatic flood detection when an appropriate water index technique is employed. In contrast to the referenced study, our research will utilize CNN and VGG16 for image processing, further contributing to the advancement of flood detection methods.

This research study titled "*Synergistic Use of Geospatial Data for Water Body Extraction from Sentinel-1 Images for Operational Flood Monitoring across Southeast Asia Using Deep Neural Networks*" (Kim et al., 2021) demonstrates the effective application of deep neural networks in maximizing the potential of Sentinel-1 data and flood-related geospatial datasets. The study focuses on customizing and optimizing the U-Net model to accurately utilize various data sources, including the digital elevation model (DEM), Slope, Aspect, Profile Curvature (PC), Topographic Wetness Index (TWI), Terrain Ruggedness Index (TRI), and Buffer, specifically for the Southeast Asia region.

To evaluate the performance and segmentation accuracy of the model, a dataset comprising 128 cases with stacked layers was employed. The initial Sentinel-1 images from Vietnam, Myanmar, and Bangladesh were used to segment water bodies, and the results were analyzed. Out of the 31 cases examined, improvements were observed in Overall Accuracy (OA), while 19 points demonstrated enhanced averaged intersection over union (IOU) and F1 scores for the three segmented Sentinel-1 images. The 'Sentinel-1 VV' band yielded average scores of 95.77 for OA, 80.35 for IOU, and 88.85 for F1. In contrast, the 'band combination VV, Slope, PC, and TRI' exhibited improved scores of 96.73 for OA, 85.42 for IOU, and 92.08 for F1. These findings highlight the benefits of leveraging geospatial data and were substantiated by the extracted results in the Chindwin River basin. The study demonstrated a 7.68 percent enhancement in deep learning-based water body extraction from Sentinel-1 images by incorporating geospatial data.

Furthermore, the newly developed water body extraction model was validated through visual inspection and an evaluation of its performance, encompassing training and inference time. This model proved to be instrumental in facilitating operational flood monitoring across the Southeast Asia region. However, it is important to note that this research primarily focuses on the assessment of satellite data and available geospatial layers. Future investigations should aim to develop more reliable water body extraction models for operational flood monitoring, incorporating additional geospatial layers as well as non-geospatial data. Addressing the misclassification of factors like roads and airports is crucial to achieve higher classification accuracy and should be thoroughly verified in future studies.

2.1.2 Flood Monitoring System through Image Processing using Convolutional Neural Network

The research study titled "*Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network*" (De Vitry et al., 2019) utilized surveillance cameras to assess flood levels by employing a deep convolutional neural network (DCNN). The study employed a two-step process, wherein floodwater segmentation was performed on individual video frames. Subsequently, these frames were summarized to evaluate the temporal variations in water depth. The researchers curated a dataset comprising 1218 labeled images obtained from the Internet, which were manually annotated as their image dataset. To capture fluctuations in the actual water level, a Static Observer Flooding Index (SOFI) was computed consecutively. By employing a DCNN and the novel SOFI index, the researchers successfully extracted flood level trend information from conventional security cameras.

The research study titled *"Flood Depth Mapping in Street Photos with Image Processing and Deep Neural Networks"* (Kharazi et al., 2021) employed image processing techniques, specifically convolutional neural networks (CNNs), as well as edge and line detection models. The primary objective was to measure flood depth by referencing it to a real size benchmark, captured from multiple angles. The study utilized the Canny edge detector and probabilistic Hough transform to detect vertical edges and identify potential pole candidates within the images. This approach was chosen due to the standardized dimensions of stop signs, where each pixel in the photograph corresponded to one inch in pole length. The computation of floodwater depth involved calculating the difference in pole length values between pre- and post-flood photos taken from the same position. Moving forward, future research initiatives in this field may focus on addressing additional sources of error in pole length estimation, such as water reflection, pole shape, backdrop effects, and image resolution.

The research study titled "*River segmentation for flood monitoring*" (Lopez-Fuentes et al., 2017) explores a flood monitoring system that utilizes three distinct Deep Learning segmentation algorithms: (1) Fully Convolutional Networks for Semantic Segmentation (FCN-8s); (2) Fully Convolutional DenseNets for Semantic Segmentation (Tiramisu); and (3) Image-to-Image Translation with Conditional Adversarial Networks (Pix2Pix) to accurately estimate the water area from river images. FCN-8s incorporates well-established classification networks like AlexNet, VGGnet, and GoogLeNet into the FCN architecture. Tiramisu adapts a CNN network, originally proposed for image classification, into an FCN network with skip and upsampling layers for performing semantic segmentation. The Pix2Pix network is trained to map input images to corresponding output images using a loss function. The dataset is split into 75% for training and 25% for testing. The results indicate that Tiramisu outperforms the other algorithms in water segmentation, demonstrating the highest mean in both metrics: MIoU (81.91%) and Pa (90.47%). Additionally, Tiramisu exhibits a lower standard deviation compared to the other algorithms: MIoU (13.74) and Pa (9.16).

Table 2.1 Comparative Matrix of Convolutional Neural Network for Flood Monitoring & Prediction

Author	Year	Title	CNN Model	Data Gathering	Accuracy	Recommendations
M. Anbarasan, BalaAnand Muthu, C.B. Sivaparthipan, Revathi Sundarasekar, Seifedine Kadry, Sujatha Krishnamoorthy, Dinesh Jackson Samuel R., and A. Antony Dasel	2020	Detection of flood disaster system based on IoT, big data and convolutional deep neural network	Convolutional Deep Neural Network (CDNN)	Four sensor values from the dam: (Water Flow sensor data, Water Level sensor value, Rain Sensor value, and Humidity)	93.23%	Enhanced with IoT based devices with even longer range of sensors with decreased cost with futuristic algorithms used in every stage of the flood detection process.
Nobuaki Kimura, Ikuo Yoshinaga, Kenji Sekijima, Issaku Azechi, and Daichi Baba	2020	Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions	Convolutional Neural Network	Hourly datasets for rainfalls and water levels from 1992 to 2017 and 2000 to 2019 from the website of the hydrology and water-quality database	88.8 – 93.5%	Lead time in the prediction should be extended from an hour to three to six hours based on the time lags for the watersheds.
Pallavi Jain, Bianca Schoen-Phelan, and Robert Ross	2020	Automatic Flood Detection in Sentinel-2 Images Using Deep Convolutional Neural Networks	Deep Convolutional Neural Network	MediaEval 2019 competition	N/A	Extend approach with flood water segmentation, to have better flood mapping along with flood detection.

Junwoo Kim, Hwisong Kim, Hyungyun Jeon, Seung-Hwan Jeong, Juyoung Song, Suresh Krishnan Palanisamy Vadivel and Duk-jin Kim *	2021	Synergistic Use of Geospatial Data for Water Body Extraction from Sentinel-1 Images for Operational Flood Monitoring across Southeast Asia Using Deep Neural Networks	Customized U-Net Model	Sentinel-1 data and flood-related geospatial datasheets	7.68 %	Derive reliable water body extraction models for operational flood monitoring and test more geospatial layers and non-geospatial data.
Matthew Moy de Vitry, Simon Kramer, Jan Dirk Wegner, and João P. Leitão	2019	Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network	Deep Convolutional Neural Network (DCNN)	Surveillance footages	75%	Assess the actual value of the information provided by SOFI for the validation and calibration of urban flood models.
Bahareh Alizadeh Kharazi and Amir H. Behzadan	2021	Flood depth mapping in street photos with image processing and deep neural networks	Mask R-CNN	Publicly available flood photos that contain at least one visible stop sign	Pre-flood (89.31%) Post-flood (94.97%)	Addressing other causes of error in pole length estimation such as water reflection, pole shape, background effects, and image resolution.
Laura Lopez-Fuentes, Claudio Rossi, and Harald Skinnemoen	2017	River segmentation for flood monitoring	Fully Convolutional DenseNets for Semantic Segmentation (Tiramisu)	Images from surveillance cameras in riverbeds	MIoU (81.91%), Pa (90.47%)	Implementation of automatic detection of water level increase using cameras in riverbeds to improve flood early warnings.

2.2 Flood Management and Forecasting System

2.2.1 Flood Monitoring, Prediction, and Alert System using Internet of Things

The present research study titled "*Flood Early Warning Detection System Prototype Based on IoT Network*" (Simatupang et al., 2019) introduces a novel prototype for detecting floods at an early stage. The system's core microcontroller, Arduino UNOR3, effectively captures input signals from the HCSR04 ultrasonic sensor. Utilizing IoT capabilities, the SIM900 GSM & GPRS module receives the data transmitted by Arduino and forwards it to a predetermined website, while also sending a notification to a preprogrammed phone number when a specific water level threshold is reached. Users can obtain real-time information regarding the water level by sending a pre-formatted message to the GSM module, which retrieves the current water level data through the GPRS module. This research aims to enhance flood detection mechanisms through the utilization of IoT technology.

In this research study titled "*The Implementation of IoT Based Flood Monitoring and Alert System*" (Shah et al., 2018) the authors present a comprehensive system designed to monitor flood levels and provide timely alerts. The project follows the waterfall model, encompassing seven distinct phases: Planning, Requirement Analysis, Design, Implementation, Testing, Maintenance, and Documentation. To achieve the objectives, the system incorporates various components, including a water sensor (SEN113104 model), Raspberry Pi, and a GSM module (USB 3G modem Huawei mobile broadband E173). The Raspberry Pi collects data from the water sensor and transmits it to the GSM module, which sends out SMS alerts to residents. The implementation of the system primarily utilizes the Python programming language. The performance evaluation involves assessing the system's response time through the introduction of time delays. Additionally, two environmental tests were conducted: the Full Volume Flow Rate environment for rapid water rise and the Half Volume Flow Rate environment for gradual water rise.

In this research study, conducted by S. Bande et al., an Internet of Things (IoT)-based flood prediction system was developed using an Artificial Neural Network (ANN). The study employed various sensors including a temperature sensor (LM35), atmospheric pressure sensor, humidity sensor (BME280), and water level sensor (FDC1004). The ANN was utilized for prediction analysis, employing two algorithms: Gradient Descent with an adaptive learning algorithm and the Levenberg-Marquardt algorithm. The flood disaster prediction system demonstrated superior real-time monitoring and prediction capabilities compared to existing approaches. While the first algorithm yielded inaccurate predictions, the Levenberg-Marquardt algorithm achieved an impressive accuracy of approximately 88 percent in rainfall prediction (Bande et al., 2017).

In this research study conducted by R. Hameed et al., a novel model was developed for early flood detection and avoidance by leveraging the capabilities of the Internet of Things (IoT). The proposed system incorporated both an ultrasonic sensor and a soil moisture sensor, while ensuring compatibility with IoT standards through the utilization of the ESP8266 Wi-Fi Module. Data acquired from these sensors was subsequently transmitted to a microcontroller for processing. The collected data was then securely transferred to the cloud, enabling visualization and access through a dedicated mobile application. Remarkably, the implemented system demonstrated a substantial enhancement in water level estimation when compared to prevailing conventional methods (Hameed et al., 2020).

The research study titled "*Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks*" (Rani et al., 2020) aims to propose an efficient and adaptable approach for flood detection and alert systems using Machine Learning techniques. The system incorporates only two types of sensors, namely rain and water flow, strategically positioned across three distinct locations. By leveraging IoT technology, this system predicts potential floods and promptly notifies nearby villages, conveying information about the imminent flood risks. Moreover, it includes the computation of the time required for the floodwaters to reach a specified area. The employed Machine Learning models encompass logistic regression, linear regression, support vector machines (SVM), and neural networks. The research utilizes two distinct datasets comprising historical rainfall data, spanning from 1951 to 2000 and 1901 to 2015, respectively. To assess model performance, the evaluation metrics employed are mean absolute error and standard deviation. Findings from the study indicate that while machine learning models may not exhibit optimal performance in rainfall prediction due to inherent rainfall fluctuations, this approach remains viable for predicting floods in any state, leveraging the available data.

In the research study titled "*Design of Information Monitoring System Flood Based Internet of Things (IoT)*" (Satria et al., 2018) the researchers employed the Software Development Life Cycle (SDLC) as their software development framework, encompassing the stages of analysis, design, implementation, testing, and maintenance. To support their investigation, they utilized various components, including the ATMEGA238 Arduino Microcontroller with 8-bit processing power and a 2KB RAM capacity. By combining the Rain Sensor, Ultrasonic Sensor, and Arduino Uno Microcontroller, they developed a prototype design for a flood monitoring system. The prototype comprised an ultrasonic sensor to monitor water levels within the pipes and a rain sensor to detect weather conditions. The Arduino Uno microcontroller, equipped with an Ethernet shield, processed the height and rainfall data, which were subsequently transmitted to a web server. Users could access the flood monitoring information system by visiting the designated website and entering the web server's IP address, which in this case was 192.168.0.4. Flood-related information, such as real-time rainfall conditions and flood maps, was generated based on the user's IP address. The real-time flood early warning system was developed using HTML and JavaScript programming, with discussions involving the embedded C language of the Arduino Uno. The flood early warning information included flood maps, real-time rain conditions, and hazard status. In line with this study, the researchers of the current study will also employ the Arduino Uno Microcontroller and leverage a website for improved data processing and collection.

2.2.2 Flood Monitoring, Prediction, and Alert System using Wireless Sensor Network

A research study titled entitled "*Experiments of an IoT Based Wireless Sensor Network for flood monitoring in Colima Mexico*" (Mendoza-Cano et al., 2021) was conducted to develop a wireless sensor network (WSN) based on IoT technology for monitoring flood-related data. The WSN employed weather station nodes that functioned as servers, equipped with Ethernet technology and ample local storage. RiverCore water level nodes utilized ultrasonic water level sensors (Toughsonic) and connected to the server through 3G wireless communication technology, incorporating local data storage capabilities. Drifters, which are floating devices with GPS and local storage, interacted with RiverDrones via the 900 MHz Long-Range RF Module (XBEE XSC PRO) technology. RiverDrones, in turn, tracked the drifters and established communication using the same 900 MHz Long-Range technology, while connecting to the central server via 3G. The devices were programmed primarily in C++. To establish connections with the central server, the IoT publish-subscribe-based messaging protocol (MQTT) was employed, with messages transmitted in JSON format to ensure content standardization.

In their research study, M. Subashini et al. introduced a Flood Monitoring System utilizing Wireless Sensor Networks (WSNs). The deployed sensors encompassed temperature, ultrasonic, level, and rain sensors. The acquired data was transmitted to the Wemos D1R2 microcontroller, which was subsequently interfaced with a cloud server. Subsequently, the data collected from the sensor nodes was compared to predefined threshold values. When the water level attains the desired threshold, an alert system was triggered through a mobile application, notifying the users. This system was designed to deliver real-time alerts specifically for flood-prone areas. Moreover, potential future enhancements were identified for this system, such as incorporating an LED display to visualize the sensor values and exploring alternative networks to ensure communication resilience during connectivity disruptions or adverse weather conditions (Subashini et al., 2021).

This research study titled entitled "*A Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks*" (Patil et al., 2019) presents a comprehensive flood warning system that addresses four essential factors: (1) data collection through gaging, (2) data processing, (3) hardware and software components, and (4) dissemination of flood warning information. The system incorporates an array of sensors including an ultrasonic sensor for water level and rainfall measurement, a BMP180 barometric sensor for air pressure measurement, a DHT11 sensor for temperature and humidity, and a Node MCU ESP 8266 microcontroller. The ESP collects data from these sensors, performs computations, and uploads the information to a server. The data is then stored in a cloud database and transmitted to the front-end of both web and mobile applications. By enhancing the accessibility of emergency evaluation, this proposed system significantly improves the effectiveness and efficiency of responding to major disasters.

The present research study titled "*An Intelligent Flood Monitoring System for Bangladesh Using Wireless Sensor Network*" (Sakib et al., 2016) introduces a flood monitoring system integrating a neuro-fuzzy controller and a wireless sensor network (WSN). The system components encompass an Arduino board, Xbee modules, Raspberry Pi, GIS map, water level measurement sensor, rainfall measurement sensor, rate of change of water level height, wind direction measurement sensor, and barometer. Sensor data is transmitted to the distributed alert center via an Arduino microcontroller and XBee transceivers. At the distributed alert center, a Raspberry Pi microprocessor and an XBee transceiver are employed to generate flood alerts based on the collected sensor data. Historical flood monitoring data spanning two decades is leveraged to estimate flood duration and is stored in a database. Additionally, an intelligent Neuro-Fuzzy Controller (NFC) is developed on the Raspberry Pi microcomputer, utilizing sensor data to dispatch flood alerts.

The research study titled "*Real-time WSN Based Early Flood Detection and Control Monitoring System*" (Thekkil et al., 2017) employed the Intel IXP420 Processor, a high-performance memory controller, in conjunction with the flexible clocking CC2430F to create a comprehensive system-on-chip solution for Zigbee wireless sensor networks operating at 24 GHz. The experimental process commenced with the acquisition of images from a CMOS camera, controlled by the Intel processor. Subsequently, these images were stored in the dynamic memory of the processor via the system bus. The captured photos were then compressed and stored in static memory before being transmitted through Zigbee. Remote monitoring, facilitated by a remote host, played a crucial role in this study. The host received photographs from flood detection zones, analyzed and archived them, and displayed the images on a screen for virtual examination. Subsequently, the host sent the images to authorized mobile users, aiding in decision-making. The study introduced a novel approach utilizing an image collection unit that captures real-time images and transmits the data to a remote host via Zigbee and GSM networks. The host then compares the received data with existing information to make informed judgments, resulting in automated warning generation. The system's data output is derived from the real-time graph of the SIFT (Scale-invariant feature transform) algorithm implemented in the NetBeans Java software-based flood monitoring and detection system. In contrast to the study, our research will employ an improved camera module to enhance image processing accuracy and utilize a distinct algorithm for data analysis.

2.2.3 Flood Monitoring, Prediction, and Alert System using Microcontrollers

The research study focuses on image processing techniques using the Raspberry Pi and noir camera module. The study emphasizes the utilization of OpenCV to capture and process stereoscopic images. A camera network design is constructed and implemented in a practical field setting. The Python binding for the OpenCV computer vision library offers diverse script samples, serving as a source of inspiration for users to develop their own customized scripts. The study explores the use of extended base lines, eliminating the need for a mechanical bar, to achieve optimal stereoscopic recording. To mitigate the presence of moving objects in the frames, a series of images are captured. Moreover, the study demonstrates the feasibility of low-effort infrared photography facilitated by the noir camera. The entire setup, including the computer, accupack, and lens board, is enclosed within a 3D printed housing (Abaya et al., 2017).

The present research study, titled "*Smart River Monitoring and Early Flood Detection System in Japan Developed with the EnOcean Long Range Sensor Technology*" (Purkovic et al., 2019), aims to enhance the monitoring and detection capabilities in response to flood events. The project primarily involves the integration of an ultrasonic sensor model MB7383 with the EnOcean module, a wireless technology powered by solar energy that facilitates efficient transmission of high data rates while minimizing energy consumption. To establish the connection between the EnOcean module and the ultrasonic sensor, a compact printed circuit board (PCB) is employed, which incorporates the STM32L071K microcontroller and a step-up converter. The communication between the sensors is governed by the Generic Sensor Interface (GSI), which specifies the communication protocols and standards.

In this research study conducted by M. Noor et al., a flood monitoring and prediction system was developed utilizing the Arduino Uno platform. The system incorporates various sensors, namely ultrasonic, light, and alarm sensors, to effectively monitor flood conditions. Three distinct scenarios are considered for generating graphs, including unchanged water levels, rising water levels, and decreasing water levels. By analyzing four sets of water level data, the researchers aim to derive a prediction level. The primary objective of this system is to collect up-to-date water level readings using Arduino and generate precise and reliable data (Noor et al., 2020).

This research study titled "*Design and Development of Flood Monitoring and Early Warning System*" (Dondang et al., 2021) presents the development of a comprehensive system comprising a rainfall sensor, temperature sensor, humidity sensor, microcontroller (Arduino UNO), and GSM modem. The primary objective of this study is to propose a cost-effective flood monitoring system that can be deployed, monitored, and evaluated at selected locations in Malaysia. The system is designed to continuously monitor rainfall rates, temperature, and humidity levels to determine the potential occurrence of floods. Collected data is transmitted to a personal computer (PC) via a GSM modem. The evaluation of the system entails several tests, including (1) assessment of the DHT-11 sensor, (2) verification of the GSM modem functionality, (3) validation of the RG-11 optical rain sensor, and (4) overall system integration and evaluation on the microcontroller. Modifications to the system's design were implemented to address the non-waterproof nature of certain components. The proposed system demonstrates the capability to measure rainfall intensity, predict the likelihood of flooding, and issue text message alerts to registered users.

The research study titled "*Flood Disaster Indicator of Water Level Monitoring System*" (Hassan et al., 2019) aims to develop an improved flood monitoring system. In this study, Arduino Uno Microcontroller and Float Switch sensors were employed to monitor flood levels. The system differentiates between escalation levels by utilizing three float switch sensors, with the Arduino Uno controller handling the data transmitted by these sensors. The controller has been programmed to accept a variety of inputs and outputs via the transmission pin, and the C compiler program is used to deliver commands using appropriate AT commands. The GSM module receives the output transmitted by the Arduino Uno controller, and the user receives an SMS alert accordingly. The SMS alerts vary depending on the detected sensor data. The GSM network is utilized to deliver data to the user via SMS alarms, which have demonstrated reliability even when phone lines are overloaded due to operating on a different frequency. Additionally, SMS offers the advantage of transmitting data to multiple users simultaneously. The flood monitoring system prototype consists of a transmitter and a receiver. The transmitter is responsible for sending and receiving data from the sensors, while the receiver obtains data from the transmitter and relays it to the device supporting the local Telco Sim. Throughout the monitoring process, the prototype is placed in a water tank, and calibration of the water level with the sensors is necessary for precise results. The monitoring procedure is conducted at various times and under different weather conditions to assess the prototype's effectiveness in diverse scenarios. In contrast to the referenced study, this research will utilize an Arduino Uno Microcontroller to enhance the processing of data collected from the sensors.

Table 2.2 Comparative Matrix of Flood Management and Forecasting System

Authors	Year	Title	Sensors	Software Algorithm	Microcontroller	Power Management	User Interface Network	Recommendations
D. Satria, S. Yana, R. Munadi, and S. Syahreza	2017	Design of Information Monitoring System Flood Based Internet of Things (IoT)	Rain Sensor, and Ultrasonic Sensor	HTML and Javascript programming integrated with C language embedded in Arduino Uno.	Arduino (ATMEGA238)	Power Supply	Website	N/A
W. H. W. Hassan, A. Z. Jidin, S. A. C. Aziz, and N. Rahim	2019	Flood disaster indicator of water level monitoring system	Float Switch sensor	C program with AT commands	Arduino Uno, and GSM Module	Power Supply	SMS	Additional button functions for their GUI, use wireless sensors, use of LED display for road street clients and security, powered by solar panel
T. M. Thekkil and N. Prabakaran	2017	Real-time WSN based early flood detection and control monitoring system	CMOS camera	SIFT (Scale-invariant feature transform) algorithm	Intel IXP420 Processor	Power Supply	Website	N/A

S. N. Sakib, T. Ane, N. Matin, and M. S. Kaiser	2016	An intelligent flood monitoring system for Bangladesh using wireless sensor network	water level measurement sensor, rainfall measurement sensor, rate of change of height of water level, wind direction measurement sensor, and barometer	NFC (Neuro-Fuzzy Controller)	Arduino, Xbee, Raspberry Pi	Power Supply	GIS Map	N/A
D. S. Rani, G. N. Jayalakshmi, and V. P. Baligar	2020	Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction	Rain drop sensor and water float sensor	logistic regression, linear regression, SVM, and neural networks	Raspberry Pi	Power Supply	LCD Display, IoT Gecko Cloud	N/A
R. A. Dondang, Z. Mansor, and G. B. Satrya,	2021	Design and Development of Flood Monitoring and Early Warning System	rainfall sensor and temperature sensor	Arduino IDE software	Arduino Uno, GSM modem	Power Supply	SMS	Add more sensors specifically pressure sensor, replace GSM with IoT
S. Patil, J. Pisal, A. Patil, S. Ingavale, P. Ayarekar, and S. Mulla	2019	Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks	ultrasonic sensor, BMP180 barometric sensor, DHT11 for temperature	Cloud Architecture, Website and Android Software	Node MCU ESP 8266	Power Supply	Website and Mobile Application	N/A
] M. J. Subashini	2021	Early Warning System using Weather Forecasting Data and Wireless Sensor	Temperature sensor, ultrasonic sensor, level sensor, and rain sensor	Long Range Communication Protocol LoRA	Wemos D1R2	Power Supply	Mobile Application	LED Display of sensor values, early alert of flood in ten days,

		Networks-A Review						alternate network for communication
W. Shah et al	2018	The Implementation of IoT Based Flood Monitoring and Alert System	water sensor, SEN113104 model	Python Language	Raspberry Pi and GSM module	Power Supply	SMS	N/A
D. Purkovic et al	2019	Smart River Monitoring and Early Flood Detection System in Japan Developed with the EnOcean Long Range Sensor Technology	ultrasonic sensor MB7383	GSI (Generic Sensor Interface),	EnOcean module, microcontroller STM32L071K	Solar Panel	Generic Sensor Interface (GSI)	N/A
L. K. S. Tolentino et al	2019	Real Time Flood Detection, Alarm and Monitoring System Using Image Processing and Multiple Linear Regression	Float switch, rain gauge, and flow rate meter sensor	Multiple Linear Regression	Arduino Uno and Raspberry Pi	Solar Panel	Mobile Application	Collect more data of the rain fall, use more compatible sensors, cooling system for the battery, improvement of algorithm
Proposed study	2022	SWAMP: Flood Monitoring, Prediction, and Alert System Using Wireless Sensor Network and Internet of Things (IoT)	Ultrasonic sensor, barometric pressure sensor, rain gauge	Time – series autoregression	NodeMCU and Raspberry Pi	Solar Panel	Website	N/A

2.3 Time series: Autoregression

2.3.1 NARX Model

This research study aims to investigate the interactions between ground and surface waters in karst limestone locations, particularly in low-lying areas. These interactions are influenced by the unique hydrogeological dynamics of the bedrock aquifer. Under extreme hydrological conditions, such interactions can lead to widespread and long-lasting floods, resulting in significant costs and disruptions. In this study [24], we compare the predictive performance of three models: a nonlinear autoregressive model with exogenous variables (NARX) based on nonlinear time-series analysis, machine learning-based near support vector regression, and a linear autoregressive model (ARX) based on linear time-series analysis. The comparison is conducted in the context of predicting groundwater flooding in a lowland karst region of Ireland.

The present research study proposes a prediction model to forecast daily rainfall in Hoa Binh, Vietnam, employing Nonlinear Autoregressive Neural Networks with external variables (NARX). Input factors considered in the investigation include temperature, wind speed, relative humidity, solar radiation, and daily rainfall, while daily rainfall serves as the output variable. Subsequently, a NARX-based model for daily rainfall prediction is developed and assessed using various metrics, including the coefficient of correlation (R), root mean squared error (RMSE), and mean absolute error (MAE). The obtained statistical values, namely R = 0.8846, RMSE = 5.3793 mm, and MAE = 3.0218 mm, indicate a significant correlation between the measured and predicted rainfall quantities.

Hence, it is reasonable to assert that the constructed model holds promise for accurate daily rainfall forecasting (Pham et al., 2020).

2.3.2 ARX Model

The research study titled "Enhancing Time Series Data through ARX and ANN Models" (Akouemo et al., 2017) focuses on mitigating the adverse effects of anomalous data in energy forecasting. Anomalous data is identified as a key contributor to inaccuracies in forecasting models. This study introduces a novel approach employing ARX and ANN models for anomaly detection and imputation in time series data. The proposed methodology utilizes hypothesis testing on the extreme residuals to assign anomalous data points. Results demonstrate the effectiveness of the proposed methods in identifying and imputing undesirable data points. The forecasting models are initially trained using unclean data and subsequently evaluated on an out-of-sample dataset containing unclean data. Upon integration of the cleaned data and retraining of the models, a significant improvement of 1.67% in mean absolute percentage errors and 32.8% in root mean square error is observed.

2.3.3 Multivariate Time Series Model

The present research study aims to explore the interdependencies among time steps and series and their impact on activities such as predicting electricity consumption, solar power output, and analyzing polyphonic piano works. To effectively address this challenge, the study proposes the utilization of recurrent neural networks (RNNs) with an attention mechanism, emphasizing the crucial role of long-term dependencies in time series data for accurate prediction. In addition, the study introduces the application of filters to extract time-invariant temporal patterns, which can be likened to a "frequency domain" transformation of the time series data. Moreover, the study presents a novel attention mechanism designed to identify the relevant time series variables for multivariate forecasting. The proposed model demonstrates significant improvements in performance across a range of real-world applications (Sahagun et al., 2017).

2.4 Sensors

2.4.1 Rain Gauge

The WH-SP-RG is a tipping-bucket rain gauge that measures the rate of precipitation and its total volume. It contains a hall-effect sensor that triggers a magnetic contact when the tip moves.

Figure 2.1 Rain Gauge

2.4.2 Barometric Pressure Sensor

Barometric Pressure Sensor using the BMP180 measures atmospheric pressure, the applied weight of air on everything. The sensor measures the force, including temperature, that may affect the atmospheric pressure and provides that information in digital output.

Figure 2.2 BMP 180 Barometric Pressure Sensor

2.4.3 Ultrasonic Sensor

The Arduino board observes its surroundings by accepting analog information from numerous sensors and may utilize them via LCDs, speakers, and other devices. The Ultrasonic Sensor determines its distance from a given object using non-contact technology. It measures by producing soundwaves that are then reflected to it. The measured distance is also provided on an LCD display.

Figure 2.3 JSN-SR04T Waterproof Ultrasonic Sensor

2.4.4 Web Digital Camera for Raspberry Pi

For image processing of the water level, a 1080P USB Camera 2MP Livestream Webcam for Raspberry Pi. It is a high-performance web digital product for real-time data transmission via USB port.

Figure 2.4 Web Digital Camera

CHAPTER 3

METHODOLOGY

The methodology holds a pivotal role in research studies, guiding its investigative journey. It encompasses the systematic approach to data collection, analysis, and interpretation. Through meticulous planning and rigor, researchers strive for trustworthy and reliable findings, establishing a robust foundation for scholarly contributions.

3.1 Research Design

This study implements a developmental and descriptive research design as the researchers will develop a flood monitoring, alert, and prediction system with a website for Barangay Frances, Calumpit, Bulacan residents. The study will be implementing a User Acceptance Testing or UAT in the community for functionality and accessibility testing of the device. The study will also use a developmental research design involving innovative flood monitoring and prediction systems from the previous project of SnapFlood integrating wireless sensor networks and artificial neural networks using time series autoregression for flood prediction.

INPUT

KNOWLEDGE REQUIREMENTS:

- Hydrological Sensors
- Flood management and disaster risk reduction
- Geographic
- Demographics
- Image Processing
- Machine Learning
- Arduino Programming
- Raspberry Pi
- Electronic Circuit Connections

HARDWARE REQUIREMENTS:

- Raspberry Pi 4B
- NodeMCU
- Ultrasonic Sensor
- Barometric Pressure Sensor
- Rain Gauge
- Web Digital Camera
- Solar Panel
- Solar Charge Controller
- Battery
- Pocket Wi-Fi

SOFTWARE REQUIREMENTS:

- Arduino IDE
- ThingSpeak
- Python using Visual Studio Code
- Blender

PROCESS

- Design and fabricate a flood monitoring and predictive device with integrated hydrological sensors using Wireless Sensor Network and Internet of Things.
- Develop an image processing node using a web-camera module and real-time data display of dam information monitoring system; develop a predictive model for water level using time series and dam information.
- Establish a hardware and software notification and alert system using an integrated siren and a user-friendly website that displays data from the integrated sensors and dam information for flood monitoring and prediction.
- Test and evaluate the functionality, reliability, efficiency, and sustainability of the flood monitoring, prediction, and alert system.

OUTPUT

SWAMP: Flood Monitoring, Prediction, and Alert System Using Wireless Sensor Network and Internet of Things (IoT)

Figure 3.1 Input Process Output (IPO) Model of the Study

Figure 3.1 shows the study's Input Process Output (IPO) Model including the requirements to create the project efficiently and functionally. It shows the hardware components the project will need and how to develop its system.

The Input section consists of preparations related to flood, the technologies required, and the processes needed to build the system. The necessary initial knowledge includes information about hydrological sensors, flood management and disaster risk reduction, geography, demographics, and image processing. The hardware requirements involve Raspberry Pi 4B, NodeMCU, Ultrasonic Sensor, Barometric Pressure Sensor, Rain Gauge, Web Digital Camera, Solar Panel, Solar Charge Controller, Battery, Pocket Wi-Fi, and Siren. The software requirement includes Arduino IDE, ThingSpeak, Python using Visual Studio Code, Blender, HTML, CSS, Javascript, XAMPP, Apache, MySQL, and phpMyAdmin.

The Process section of the IPO Model includes the design and fabrication of the device with integrated hydrological sensors using Wireless Sensor Network and Internet of Things, development of image processing using web-camera module and real-time display of dam information, development of predictive model using time series autoregression, establishment of alert and notification system through hardware & software, and testing the functionality, reliability, efficiency, and sustainability of the flood monitoring, prediction, and alert system.

Finally, the output of the model is the fully functioning device entitled SWAMP: Flood Monitoring, Prediction, and Alert System Using Wireless Sensor Network and Internet of Things (IoT). Once a problem occurs in the system, troubleshooting, maintenance, and repairing will be implemented to assure the sustainability and reliability of the device.

3.2 Research Process Flow

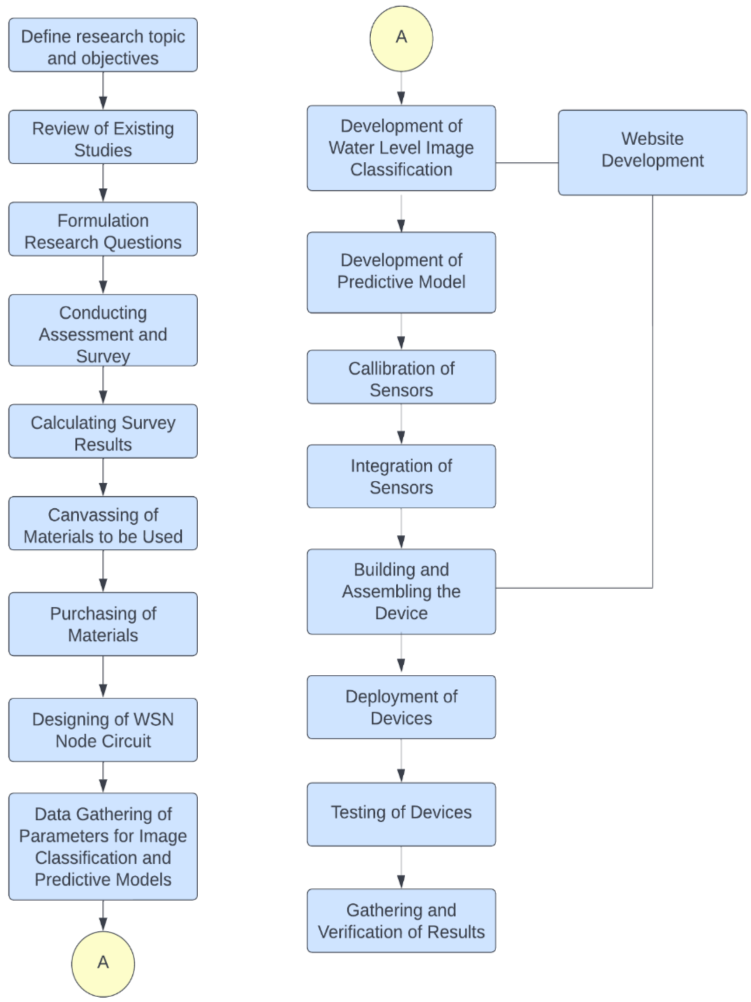


Figure 3.2 Research Process Flow for System Waterline and Monitoring Probe

The preceding diagram depicts SWAMP's research process flow. It begins by defining the research topic and its objectives, followed by a review of previous studies related to SWAMP. Following the formulation of research questions and the execution of a community needs assessment and survey, the proponents will be able to construct SWAMP in a manner that is more suitable for the residents of Barangay Frances; the results will then be evaluated and quantified. The proponents then plan to perform a canvassing and purchase the necessary building supplies for SWAMP. The designing phase will then begin, along with the data collection in which the parameters required for image classification and predictive models will be obtained. Following the collection of data, water level image classification and SWAMP's predictive models will be developed. As the backend of SWAMP is being developed, sensors will be calibrated and integrated before the device is constructed and assembled (considered as one node). The website is also being developed concurrently with the device's backend development and construction. Then, the proponents will be able to deploy and test the device(s) for data collection and verification.

3.3 Block Diagram of the Study

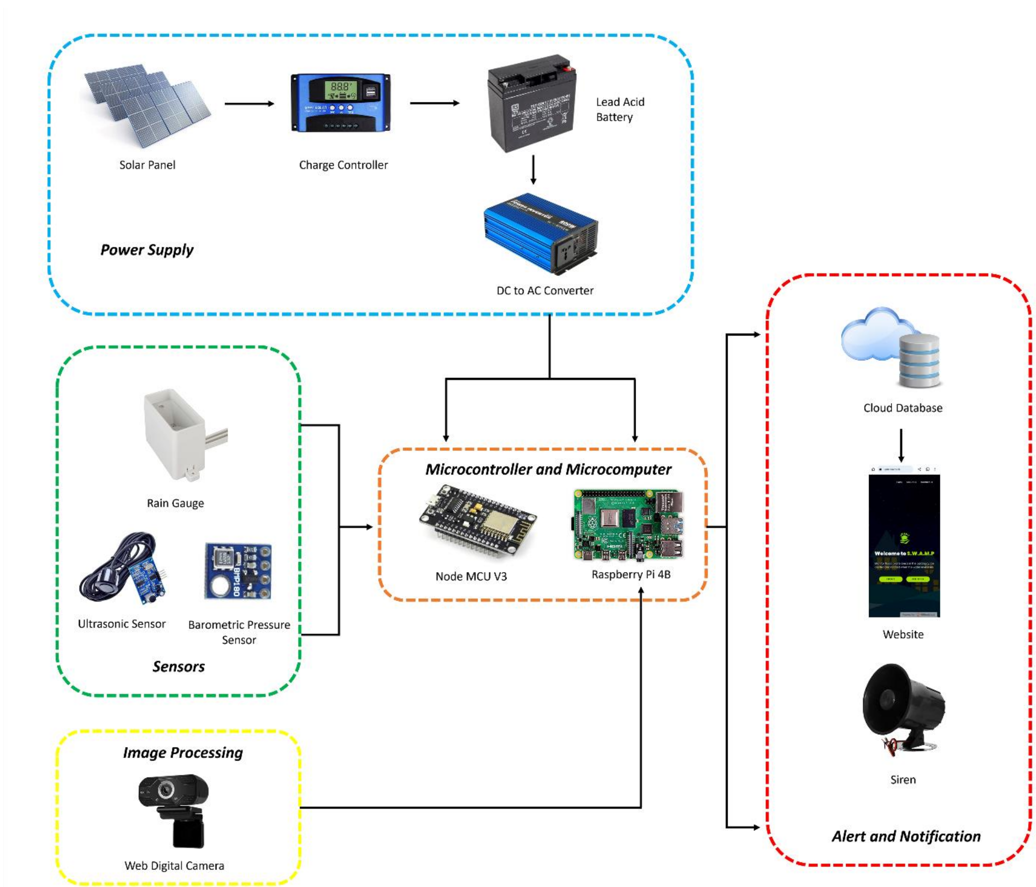


Figure 3.3 Block Diagram for SWAMP

The block diagram displays the proposed primary device depicted above. The block that contains the sensors send analog signals to the NodeMCU V3 and are then passed to the Raspberry Pi to be analyzed. The camera will capture pictures of water level measurements to send to the Raspberry Pi 4B for image processing. The block containing the microcontroller and microcomputer receives the analog signals sent from the sensors and the image captured by the camera. When the NodeMCU V3 finishes processing the input data, it will send its output to the Raspberry Pi and the image from the camera. After image processing, Raspberry Pi 4B with a predictive model using time series autoregression and cloud database will generate the result through the internet with an internet module and a notification regarding the intensity of the rainfall, the dam water spill, and the predicted flood level within 3 hours. A siren will alert the barangay residents to prepare for a flood depending on the expected occurrence of the flood. A solar panel connected to a solar charge controller and a battery will power the flood monitoring system to reduce electrical consumption and create an independent supply. A DC to AC inverter will regulate the input voltage into a certain amount that supplies the internet module, NodeMCU V3, and Raspberry Pi.

3.4 Design and fabrication of flood monitoring and predictive device using hydrological sensors

3.4.1 Geographic Mapping of Barangay Frances in Calumpit, Bulacan

The physical characteristic of the Earth is defined by geography. Flood as one of the geological processes is influenced by different natural existing factors and one of them is land elevation that can be determined by geography. Flood monitoring, prediction and alert systems can be obtained by identifying the most efficient areas to place the devices. Figure 3.4 shows the target deployment areas in Barangay Frances in Calumpit, Bulacan.

Figure 3.4 Geographic Mapping of Barangay Frances, Calumpit, Bulacan

Figure 3.4 shows how the nodes will be placed in the vicinity of the barangay. Node 1, as the parent node, is placed in the middle of Purok 5, whilst Nodes 2 and 3 are placed approximately 200 meters away from Node 1.

3.4.2 Materials and Equipment

SWAMP requires multiple sensors and single board computers to function properly. Listed below are the components that will be utilized in the development of a flood monitoring system using integrated hydrological sensors.

3.4.2.1 Power Management

SWAMP consists of a sustainable power source. Specifically, the lead-acid battery is interconnected with the solar charge controller within the system. The primary function of the solar charge controller is to effectively regulate the current and voltage that are supplied to the battery originating from the solar panel. Moreover, the USB hub serves as a connection point for the Node MCU microcontrollers, pocket Wi-Fi, Raspberry Pi, siren, and web digital camera.

Figure 3.5 Lead-Acid Battery

Lead acid batteries, known for their robustness and affordability, are widely used in applications such as automotive vehicles, uninterruptible power supplies (UPS), and renewable energy systems, providing reliable energy storage solutions.

Figure 3.6 Solar Panel

The solar panel serves as the main power source of the system. It is a renewable energy that stores a direct current and charges the battery while being regulated by the solar charge controller. The main node is powered by an 80W solar panel, while the other 2 nodes are powered by a 30W solar panel.

Figure 3.7 Solar Charge Controller

A solar charge controller, an essential component in solar power systems, regulates and optimizes the charging process of batteries, ensuring efficient energy conversion from solar panels and safeguarding the battery's lifespan by preventing overcharging or excessive discharge.

Table 3.1 List and Specifications of Materials & Equipment under Power Management

Component	Specifications
Battery (Lead-Acid)	• Battery Type: Rechargeable • Capacity: 9 Ah (20 hours) • Max discharge current: 130A • Input Voltage: 12V • Optimal Life: 5 years • Size: 151mm x 65mm x 100mm
Solar Panel	• Max Power at STC: 200W • Open Circuit Voltage: 27V • Short Circuit Current: 9.66A • Optimum Operating Voltage: 22.6 V • Optimum Operating Current: 8.85 A • Operating Temperature: -40°F~194°F / -40℃~90℃ • Maximum System Voltage: 600 VDC UL • Maximum Series Fuse Rating: 15A • Output Cable: 14 AWG 2.95 ft (0.9m) long • Dimensions: 64.96 x 26.38 x 1.38 in / 1650 x 670 x 35mm • Weight: 26.46 lb (12 kg) • Solar Cells: 40 Cells (4*10) • Junction Box: IP65 • Connectors: Solar Connectors • Module Efficiency: 18.1% • Cell dimension: (158.75 x 158.75) mm / (6.25 x 6.25) in x 40 cells
Solar Charge Controller	• Voltage: 12V/24V • Charge Current: 60A • Discharge Current: 60A • Charge Mode: PWM • Electricity Display: Yes • Light Control: Yes • Discharge Cut-off Voltage: 10.7V (default, adjustable) • Discharge Recovery Voltage: 12.6V (default, adjustable) • USB Output: 5V, 1A • Standby Current: <10mA • Temperature: -30℃ +60℃ • Size/Weight: 183mm x 90mm 47mm/290g

Table 3.1 shows the specifications of the solar panel, lead-acid battery and the solar charge controller used in the SWAMP devices.

3.4.2.2 Microcontrollers and Microcomputer

The Node MCU microcontroller processes the codes needed for the sensors to perform their functions. It also sends the data collected from the sensors to the cloud platform which will be processed through the Raspberry Pi microcomputer. The Raspberry Pi provides a cost-effective platform performing complex image analysis and manipulation tasks in various applications. It also performs the tasks needed for the predictive model.

Figure 3.8 Node MCU Microcontroller

As shown in Figure 3.8, the Node MCU is a microcontroller that processes sensor operations, and it allows communication between devices over the internet. It sends sensor data to the cloud. Its compact size, low power consumption, and support for multiple communication protocols make the NodeMCU an excellent choice for building scalable and interconnected IoT systems.

Figure 3.9 Raspberry Pi 4B

The Raspberry Pi 4B, with its powerful quad-core processor, increased RAM, and improved connectivity options, finds versatile applications ranging from home automation and media centers to robotics and edge computing, empowering enthusiasts and professionals to easily develop innovative projects.

Table 3.2 List and Specifications of Materials & Equipment under Microcontroller and Microcomputer.

Component	**Specifications**
NodeMCU	● Microcontroller: ESP8266 ● Operating Voltage: 3.3V ● Flash Memory: 4MB ● GPIO Pins: 11 ● Core Speed: 80 MHz
Raspberry Pi 4B	● Quad core 64-bit ARM-Cortex A72 running at 1.5GHz ● 1, 2 and 4 Gigabyte LPDDR4 RAM options ● H.265 (HEVC) hardware decode (up to 4Kp60) ● H.264 hardware decodes (up to 1080p60) ● VideoCore VI 3D Graphics ● Supports dual HDMI display output up to 4Kp60 ● ARMv8 Instruction Set ● Mature Linux software stack ● Actively developed and maintained

As shown in Table 3.2, the summary of specifications of the Node MCU microcontroller and Raspberry Pi is listed.

3.4.2.3 Hydrological Sensors and Web Digital Camera

The hydrological sensors used in the system are the ultrasonic sensor, barometric pressure sensor, and the rain gauge. A web digital camera is also used in the system for live-streaming and real-time monitoring of flood level.

Figure 3.10 Ultrasonic Sensor

SWAMP integrates a JSN-SR04T ultrasonic sensor to measure the water level. It can measure up to 6 meter of water level data. Median Filter was used to filter out noise produced by the sensor data.

Figure 3.11 BMP 180

The barometric pressure sensor, utilizing its precise measurement capabilities, allows for accurate monitoring of atmospheric pressure changes, enabling weather forecasting, altitude tracking, and enhancing the functionality of various applications including aviation, outdoor sports, and environmental monitoring systems.

Figure 3.12 Rain Gauge

The tipping bucket rain gauge, a commonly used precipitation measurement device, operates by collecting rainfall in a bucket that tips when a specific amount of water is accumulated, allowing for accurate and automated data collection in meteorological monitoring systems. The reed switch inside the rain gauge is replaced by a hall-effect sensor which is more reliable and can provide better results.

Figure 3.12 Web Digital Camera

A web digital camera is a device that captures high-quality images and videos, specifically designed for seamless integration with web-based applications and platforms, facilitating video conferencing and live streaming.

Table 3.3 List and Specifications of Materials & Equipment under the hydrological sensors and web digital camera.

Component	Specifications
Ultrasonic Sensor (JSN-SR04T)	● Working Voltage: DC 3-5V ● Working Current: 8mA ● Working Frequency: 40Hz ● Max Range: 6m ● Min Range: 20cm ● Measuring Angle: 75 degrees ● Trigger Input Signal: 10µS TTL pulse ● Echo Output Signal Input TTL lever signal and the range in proportion ● Dimension 42mm x 29mm x 12mm
Atmospheric Pressure (BMP 180)	● Input Voltage: 3 to 5VDC ● Logic Voltage: 3 to 5V compliant ● Pressure Sensing Range: 300-1100 hPa (9000m to -500m above sea level) ● Resolution: Up to 0.03hPa / 0.25m ● Operational Range & Accuracy: -40°C to +85°C, +-2°C temperature accuracy ● I2C Address: 7-bit address 0x77.

Rain Gauge Tipping Bucket	● Model: WH-SP-RG ● Output: pulse ● One pulse: 0.3mm
Web Digital Camera	● High Resolution COMS color sensors ● 640x480/800x600 capture size ● Video mode: 24 bit true color ● Interface: USB port ● Noise-signal ratio: greater than 48 dB ● Dynamic range: greater than 72 dB ● Image focus: 5 cm to infinity ● Build-in image compression ● Automatic Brightness Adjustment ● Automatic color compensation

Table 3.3 indicates the summary of specifications of the sensors and camera used in the system. These specifications are key factor to determine the capacity of the sensor depending on the environment, and to calibrate it properly based on its specification.

3.4.3 Circuit, 3D, and CAD Design

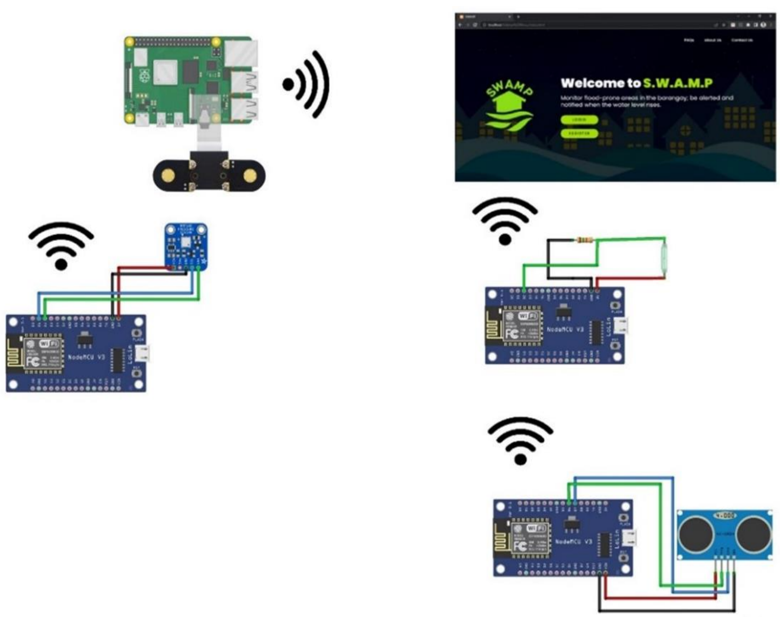


Figure 3.13 SWAMP Sensor Connections to NodeMCU, Raspberry Pi 4B, and Website

As seen on figure 3.13, the sensors used in this project will be directly connected to their respective NodeMCU. The data gathered in each NodeMCU will then be transmitted by the ESP 8266 to the Raspberry Pi 4B for Data Processing. The processed data will be used for flood monitoring and prediction with the use of image processing (via camera module) and multiple time series predictive models. The final monitored data and predicted data will then be sent to the SWAMP website for the freedom of information of the residents of Barangay Frances in Calumpit, Bulacan.

Figure 3.14 Overall View of One SWAMP Prediction, Monitoring and Alert Node

The main base station of SWAMP is composed of an outdoor panel box, sensors, siren, and solar panel attached to a 4-meter-high metal pole. To ensure the stability and safety of the attached components, the modular pole has a 2-inch diameter thickness. The panel box is attached 2 meters high above ground level.

The web-camera is attached 2.5 meters high above ground level. A 1-meter platform is extended and attached 3 meters high above ground level to secure the placement of the sensors. Also, an 80 Watts solar panel is attached 4 meters above ground level.

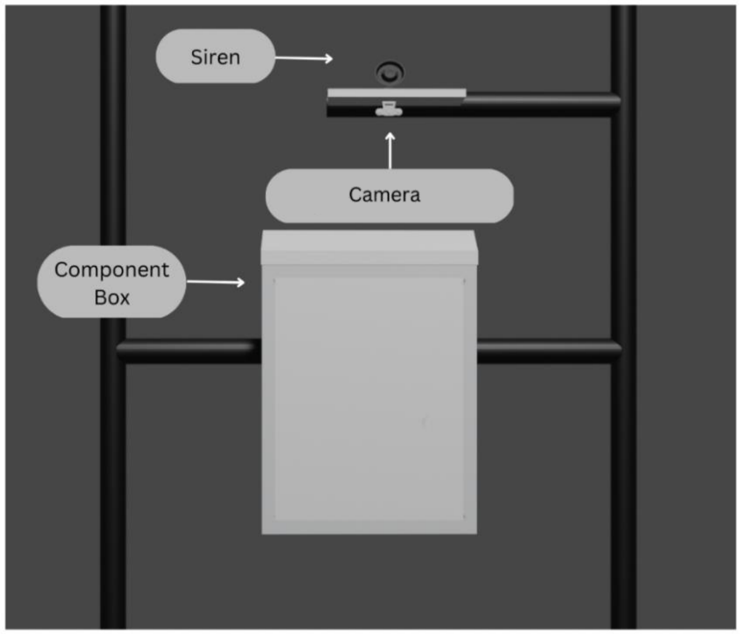


Figure 3.15 Main Components of One SWAMP Prediction, Monitoring and Alert Node

One SWAMP Node consists of essential components, including a Web Digital Camera, a siren, and a component box with microcontrollers, a battery, a solar charge controller, and a pocket Wi-Fi. The Web Digital Camera captures real-time images and video footage of flood-affected areas, aiding in flood monitoring and prediction. The siren provides audible warnings to alert residents and authorities about flood events. The component box houses microcontrollers for data processing, a battery and solar charge controller for reliable power, and a pocket Wi-Fi for internet connectivity, ensuring efficient operation of the node and facilitating real-time data transmission and access to flood information and alerts.

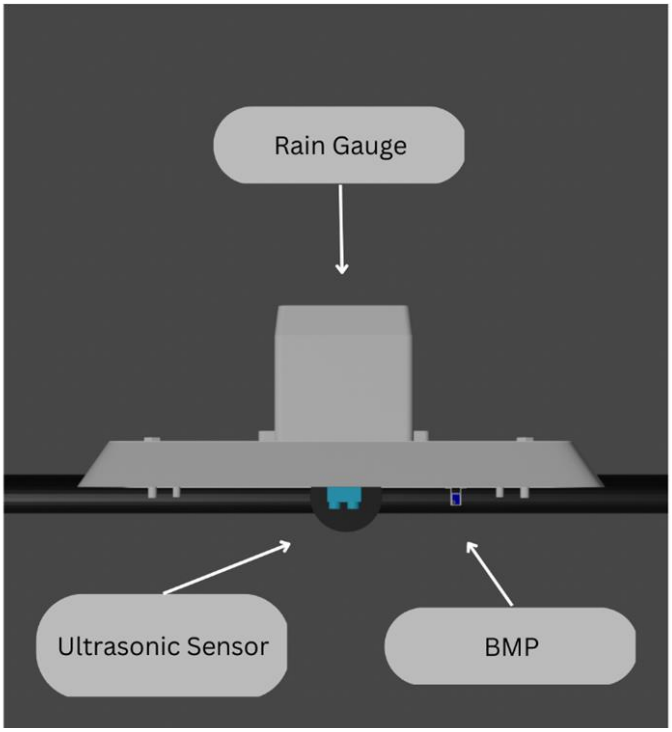


Figure 3.16 Sensor Platform in One SWAMP Prediction, Monitoring, and Alert Node

The Sensor Platform for one SWAMP Node consists of three essential components: an ultrasonic sensor for real-time water level measurement, a rain gauge for monitoring precipitation, and a barometric pressure sensor for detecting weather changes. These sensors work together to gather crucial data for flood prediction, monitoring, and alert purposes, enhancing the effectiveness of the SWAMP system in ensuring the safety of residents in flood-prone areas.

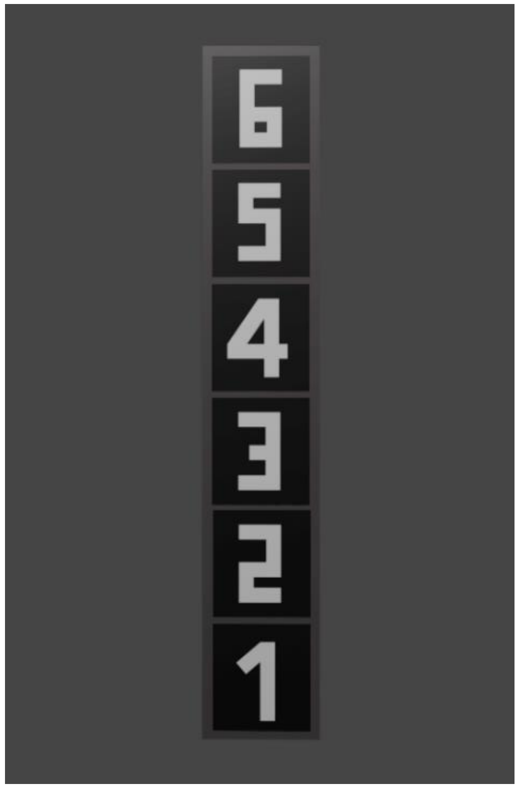


Figure 3.17 Front View of the Water Level Scaler

The figure above shows the 3D modelling of the water level scaler that will be captured by the web-camera module for the image processing program. Each number represents the measurement in feet, referenced to the local barangay's water level standard for preparation to evacuate.

3.4.4 Calibration of Sensors

Sensor calibration ensures that they are performing at their highest level and delivering the best outcomes. Calibrating is the process of comparing an instrument's accuracy to preset standards.

- **JSN-SR04T**

  Connect the ultrasonic sensor to the NodeMCU V3. The JSN-SR04T has four pins which should be connected as follows; Ultrasonic VCC to NodeMCU V3 VIN; Ultrasonic GND to NodeMCU V3 GND; Ultrasonic Echo to NodeMCU V3 digital pin 7 and Ultrasonic Trig to NodeMCU V3 digital pin 6. To print the duration of your ultrasonic sensor's pulses, upload the calibration code. The period of each pulse is another phrase for this duration. The period is the time between sending and receiving a pulse by the sensor.

- **BMP 180**

  The BMP180 communicates with the NodeMCU V3 over I2C. The BMP180 I2C pins (SDA and SCL) should be connected to D1 and D2 pin of NodeMCU V3respectively. To print the barometric pressure to the serial monitor, upload the calibration code to the NodeMCU V3 using Arduino IDE.

- **YF-S401**

There are three wires in total on this sensor: red, black, and yellow. The NodeMCU V3 VIN will be linked to the red wire, the ground to the Black wire, and the signal wire to the NodeMCU V3 digital pin 2, which is the interrupt pin. Utilizing the Hardware Interrupt 0 on the NodeMCU V3 to count the pulses. Open the Serial Monitor while the NodeMCU V3 is connected to the laptop to see the liquid volume.

- **Rain Gauge (H) WH-SP-RG**

To calibrate the Rain Gauge, connect digital pin 2 of NodeMCU to one end of the 15k ohm resistor. Attach the other end of the resistor to the Ground pin (GND). Link one end of the jack to the digital pin #2 of NodeMCU. Connect the red wire to the VIN of NodeMCU. Upload the sketch using Arduino IDE.

3.4.5 Flowchart of Integration of Sensors

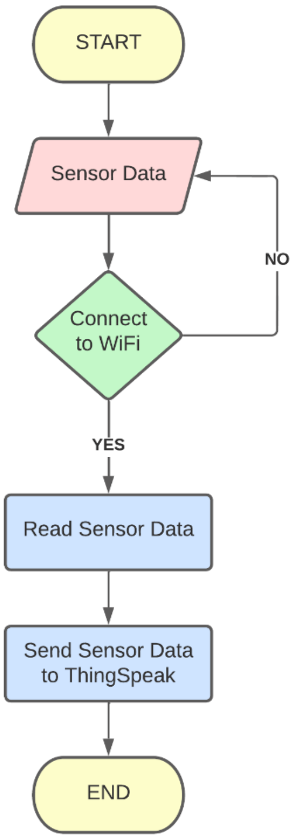


Figure 3.18 Flowchart of Integration of Sensors

Sensor Data Flowchart can be seen that when the system starts, the data input is carried out, the sensor will send data, if not, then the data input is repeated. If yes, then it is processed using IoT (Internet of Things). The data released is processed or sent to the Thingspeak Website as output on this system. The output display can be seen as a graph and the compiled sensor data can be downloaded and compressed in a csv file.

3.4.6 Integration of Sensors

The integration of multiple sensors as the foundation of reliable system that make sense of imperfect input regardless of the environment in which they operate.

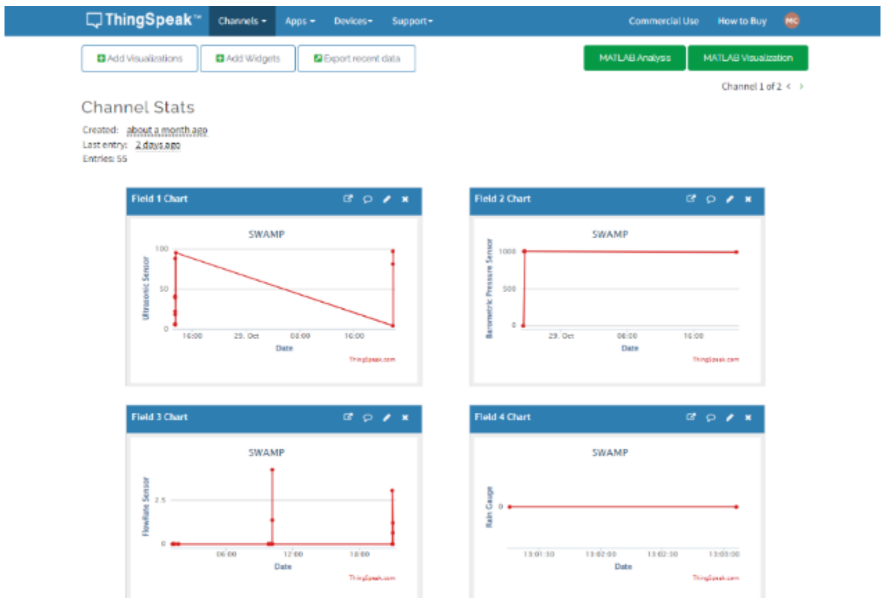


Figure 3.19 Integration of Sensors

With the use of IoT analytics platform service ThingSpeak, the gathering, visualization, and examination real-time sensor data streams online. From the sensor to the ESP8266 Wi-Fi module embedded to the NodeMCU microcontroller, sending the sensor data to ThingSpeak, visualize live data instantly, and issue notifications. Through the proper codes/sketches uploaded to the NodeMCU using Arduino IDE, 4 out of 4 sensors are successfully integrated. This platform can give the needed data for the predictive model of the system.

3.5 Development of Image Processing Using Web-Camera and Time Series Auto-Regression for Prediction with Dam Information of Flood Level

3.5.1 Image Processing of Water Level for Flood Monitoring

3.5.1.1 Flowchart of Image Processing

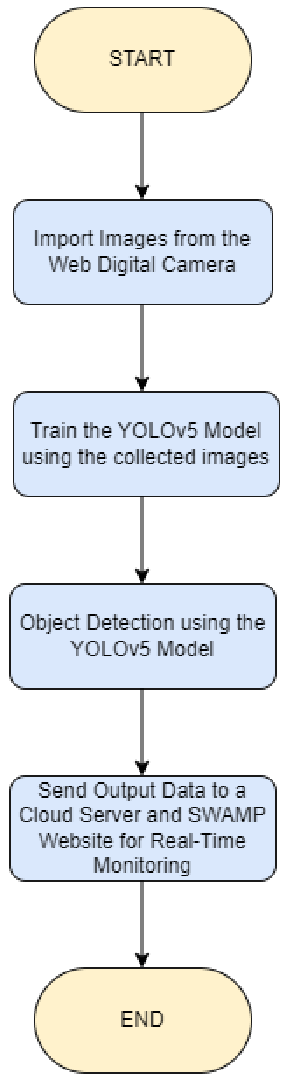


Figure 3.20 Flowchart of Image Processing

The flowchart above represents the overall process of the image processing program using python. To start the codes, a web digital camera module connected to the Raspberry Pi will scan a water level gauge integrated into the hardware aspect of the project. After the camera imports the images to the microcomputer, the program will pre-process data to convert the images to array values for the training aspect of the YOLOv5 model. The YOLOv5 model will process the trained data to classify the number to send its labeled output to the project's website for real-time water level display.

3.5.1.2 Image Processing of Water Level for Flood Monitoring

The first step in preprocessing data is to scale data in a channel with the loaded data of the program. The starting codes use a map function to provide faster access to the loaded data from the disk; a lambda function returns the images to binary values and their labels. After scaling the data between zero and one for the program to recognize it better, the next step is to shuffle the data to increase the learning capacity of the program. The second step is to split the data into two parts: training and validating the scaled data. Recognizing the length of the data is vital in partitioning them to divide the data per batch and use the split data for the learning process. Seventy percent of the scaled data belongs to the training process, thirty percent is for validation. An int function converts data into integers and uses take and skip functions will apply to the program. For each variable, the take function obtains the percentage of the data assigned to the variable, and the skip function ignores other data from its specified variable to execute the division of the scaled data.

For image processing, the model used is YOLOv5 from Ultralytics. The program creates a deep convolutional neural network that generates a specific number of neurons with their parameters: their weights and biases. The data undergoes max-pooling2D, which takes the maximum value of a region in the layer. The flatten part of the layer regularizes the processed values of the neuron in the layer for the dense layer to output a singular value. The three layers of the program use both convolution2D and maxpooling2D functions to extract the fitting data for the flatten layer to parallel with the dense layer to provide the output. Each layer adopts the ReLU activation function, and the last dense layer applies the sigmoid activation function. After constructing the deep learning model, the scaled data will use its function for training the model. The epochs will serve as the specific number of iterations of data in one cycle using ten epochs. The model displays the loss, accuracy, validation loss, and validation accuracy of the image processing learning model. This process will then proceed to test the dataset through evaluation using the test component of the scaled data. After finalizing the program, real-time footage from the web digital camera will utilize the program to assess its accuracy and behavior.

3.5.2 Flood Level Prediction Using Time Series Auto-R egression and Dam Information

3.5.2.1 Flowchart of the Predictive Algorithm

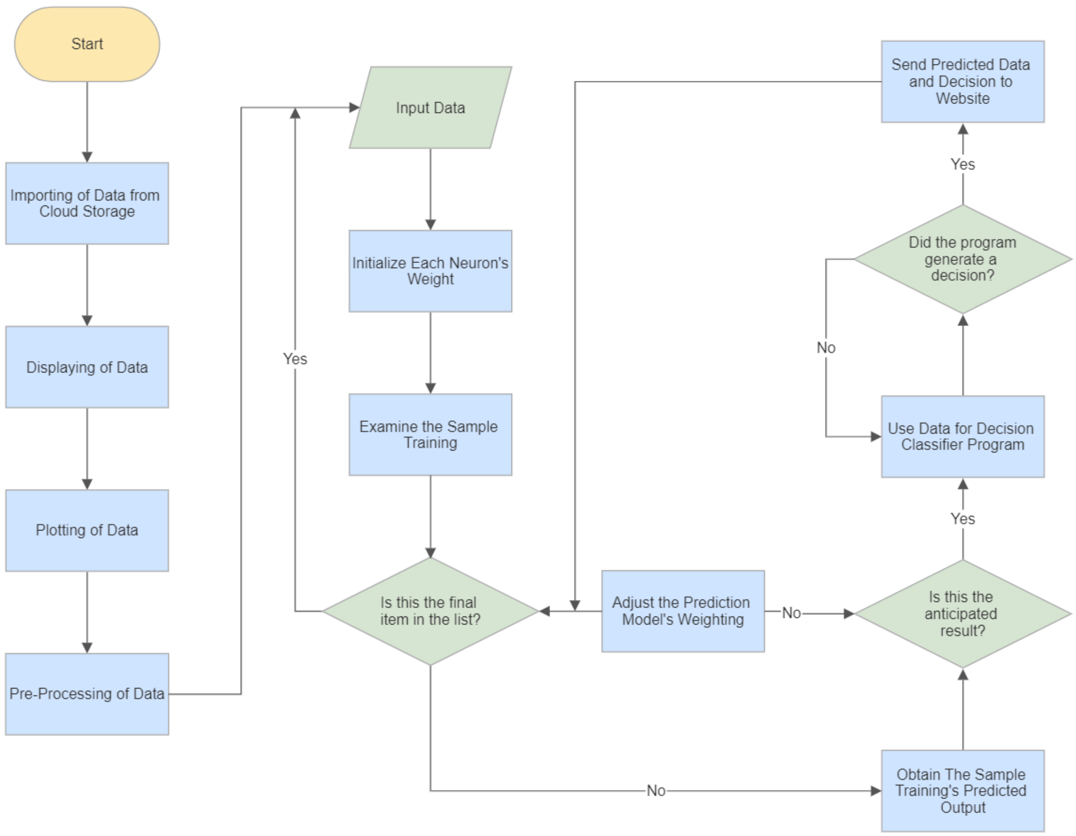


Figure 3.21 Flowchart of the Predictive Algorithm

The flowchart that the SWAMP prediction algorithm will follow is depicted in the figure above. It begins by importing, presenting, and graphing the necessary data for flood prediction in preparation for data preprocessing while utilizing python as the program language. The preprocessed data will then be utilized as input data for training, validating, and testing of predictive models. The input data will initiate the initialization of each neuron's weights, and a sample will be obtained from the input data. After collecting the required sample, the computer will determine whether the weighting should be adjusted, or the projected data should be transmitted. The decision will then motivate the algorithm to acquire further data for the continuation of prediction.

3.5.2.2 Predictive Models of Time Series Autoregression

Time series models such as Long Short-Term Memory, Gated Recurrent Unit, and 1D Convolutional Neural Network will be utilized in building SWAMP's monitoring and predictive features. This model predicts subsequent behavior based on the subject's past activities. Forecasting is enhanced when there is a correlation between the values in a time series and the values that occur before and after them.

3.5.2.2.1 Long Short-Term Memory

An extension of RNN called a Long Short-Term Memory (LSTM) network is a sequential network that maintains information and can resolve the gradient-vanishing problem encountered by RNN. At greater levels, LSTM functions similarly to an RNN cell and is consisted of three major components: a "forget gate," an "input gate," and an "output gate." While the training model remains unaltered, LSTM was developed to combat the vanishing gradient problem. LSTMs can handle noise, distributed representations, and continuous values in addition to bridging lengthy time gaps in some problems.

3.5.2.2.2 Leaky Rectified Linear Unit (Leaky ReLu)

The Leaky Rectified Linear Unit, also known as Leaky ReLU, is a variation of the standard ReLU activation function that incorporates a slight tilt in the negative range rather than a complete flat slope. The angle of the tilt, referred to as the slope coefficient, is established prior to training and remains constant throughout the training process, rather than being adapted through the learning process. The utilization of Leaky ReLU offers two advantages, namely resolving the issue of "dying ReLU" as it lacks segments with zero slope and accelerating the training process. Research has indicated that a closer proximity of the "mean activation" to 0 results in faster training.

3.5.2.2.3 Dropout

Dropout is a method in which some neurons are randomly ignored during the training process. This results in the temporary removal of their contribution to the activation of other neurons, and weight updates are not applied to the ignored neuron. In a neural network, neurons learn to specialize in certain features. However, this specialization can make the network too dependent on its context and result in a fragile model. Dropout helps to prevent this by forcing other neurons to take over the representation of the ignored neurons, leading to the creation of multiple independent internal representations. The outcome is a network that is less sensitive to the specific weights of neurons and better able to generalize and avoid overfitting.

3.5.2.2.4 Dense

The Dense layer is an essential component in neural networks as it serves as the backbone for the progression of information within the network. This layer takes all outputs generated from the preceding layer and distributes them evenly among its individual neurons. Each neuron within the Dense layer is then responsible for providing a single output that will be fed into the next layer. In essence, the Dense layer plays a crucial role in establishing the foundation for a successful neural network.

3.6 Implementation of a Siren and User-friendly Website with Sensors' Data and Dam Information Display and Web-based Notification as an Alert System

3.6.1 Front-End design and development

The proponents of the study created a website that would make Barangay Frances in Calumpit, Bulacan people' access to information easier. The platform was developed by utilizing different programming languages to create the interface and web servers to host the website on the internet. Aside from showing the monitored and predicted data, other features of the website are added, flood forecasts, flood level monitoring, water dam release, routing, flood safety rules, emergency contacts, and details about the proponents.

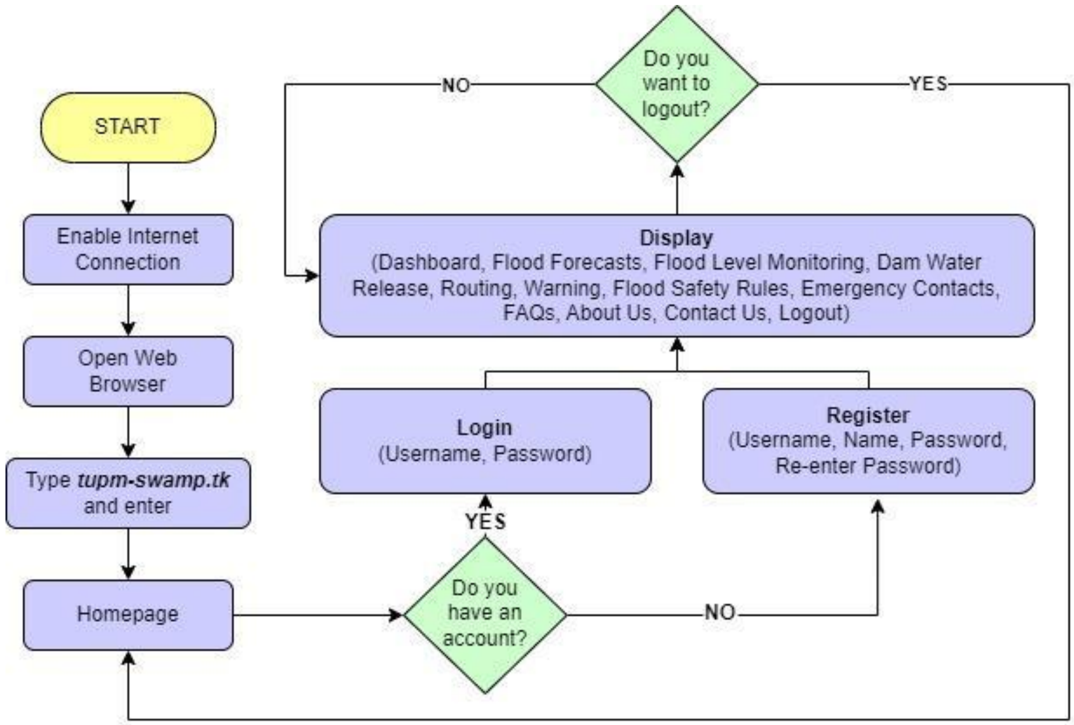


Figure 3.21 Process Flow for Accessing the Website

Figure 3.21 shows the process flow for accessing the website online. It begins by enabling the user's internet connection. An internet connection is required to fully navigate and access the website, followed by opening the preferred browser of the user and entering the URL link of the website online. This will redirect them to the homepage of the website, where two options—login and register—will be shown. The user is required to register first to create an account, and this will then be redirected to the main pages of the website showing the monitored and predicted data. The login button, however, will also redirect the user to the main pages of the website if they already have an account. The logout button can be found inside, and when clicked, the user will be redirected back to the homepage.

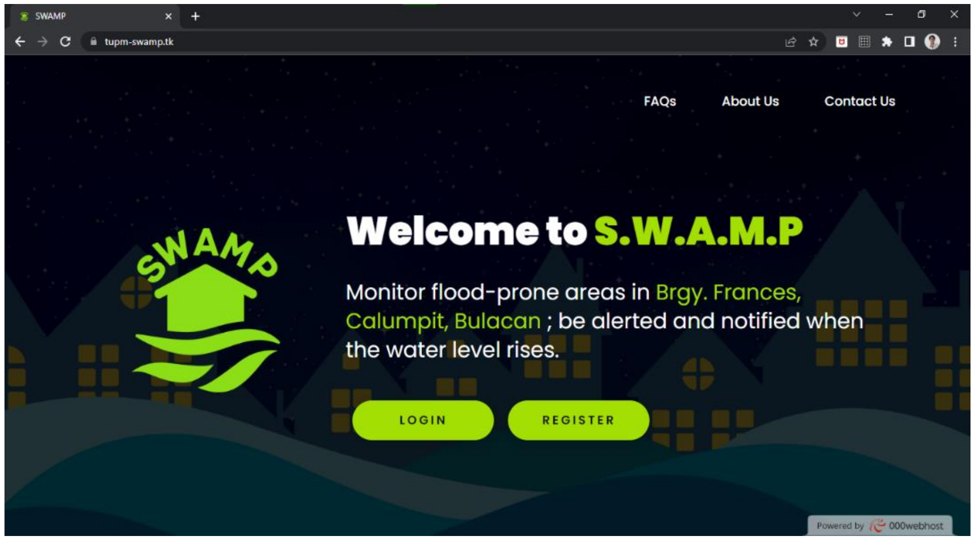


Figure 3.22 Homepage

Figure 3.22 shows the homepage of the SWAMP website, which serves as the main page for the users accessing our site. It features the website's logo and at the top is a navigation menu of subcategories for information about the website and the proponents. It has two buttons that functions as the Login and Registration of the users which is needed for them to access information about flood occurrence.

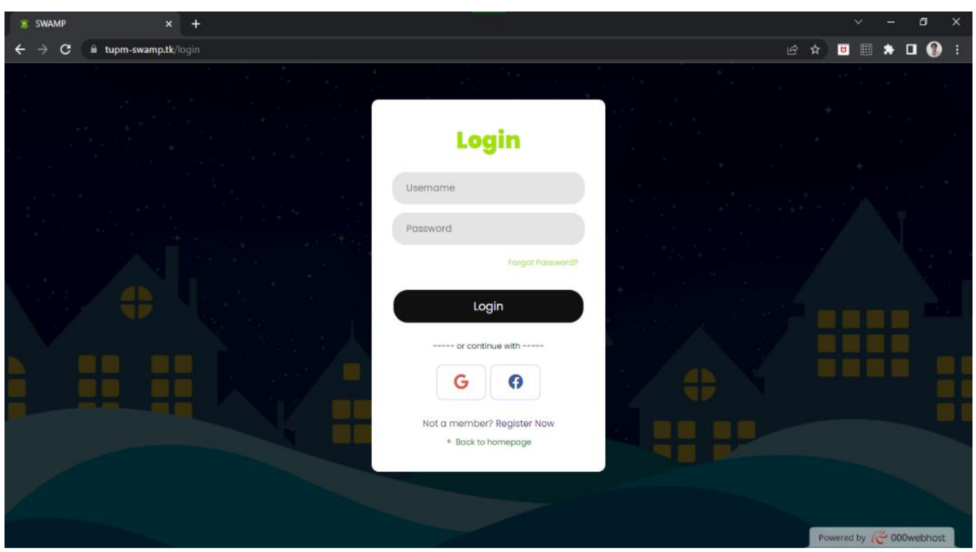


Figure 3.23 Login Page

Figure 3.23 is the Login Page of the SWAMP website. This page serves as the initial gateway in accessing the website's features and information. It consists of input fields such as the Username and for the Password, allowing the users to enter their credentials. A Login button is positioned at the bottom, enabling the users to gain access to their accounts and be redirected to the Dashboard page of the SWAMP website.

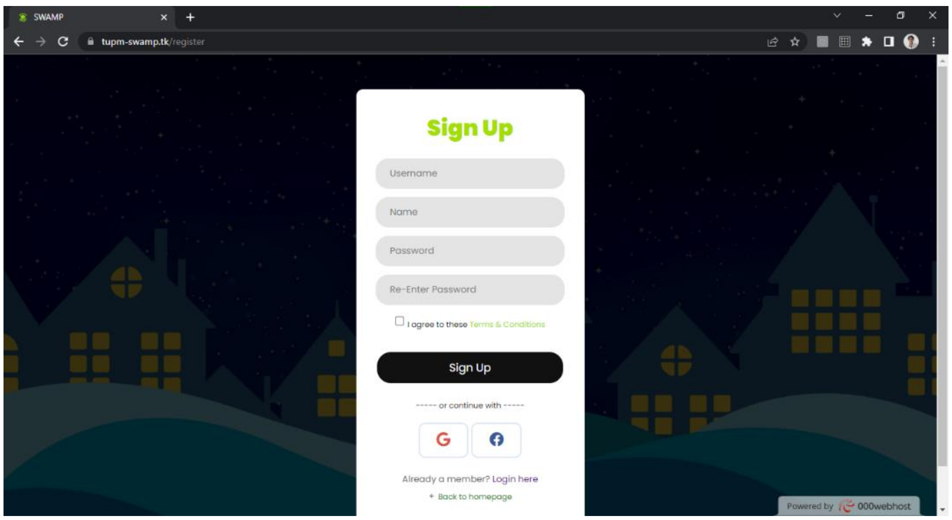


Figure 3.24 Registration Page

Figure 3.24 is the Registration/Sign Up Page of the SWAMP website. This page is designed to facilitate the users in the registration process. It has input fields that are needed to be filled up with credentials of the users to be able to create their own data on our database. A Sign-Up button is located below and will redirect to the dashboard page when clicked.

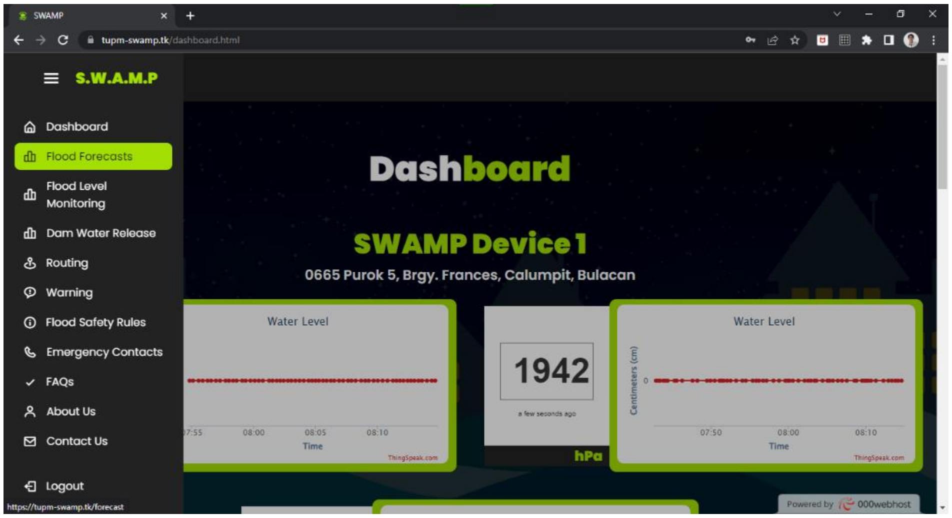


Figure 3.25 Inside Page of the Website

Figure 3.25 depicts the Dashboard Page of the SWAMP website. This serves as the central page for information regarding flood data and flood occurrence. It consists of navigation menu located under the burger button and consists of other categories of the website. It also shows a real time data in numerical values and graphs of the sensors used.

3.6.1.1 HTML, CSS, and JavaScript

The front-end of the website is the part where our users which are the people of Barangay Frances in Calumpit, Bulacan directly interacts with. The design of the interface was developed with the combination of HTML, CSS, and JavaScript codes. HTML is a markup language and CSS is a styling language. JavaScript, however, is a programming language. They are all web languages but perform different jobs on the creation of a website interface.

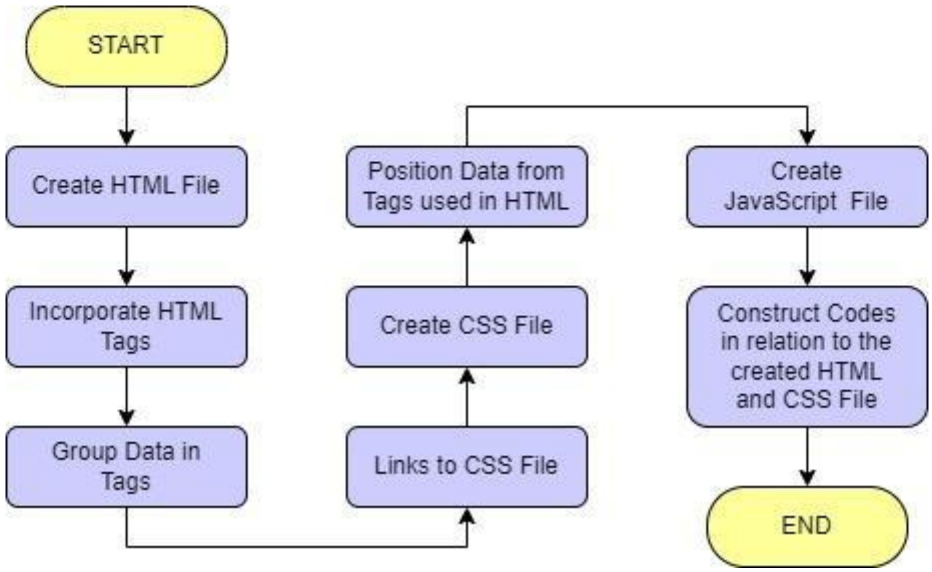


Figure 3.26 Process Flow for Designing the Interface

Figure 3.26 shows the process flow in designing the interface of the website in a combination of HTML, CSS, and JavaScript. To create the interface, an HTML file must first be created; this will handle the text and inputs of the website. In coding HTML, first incorporating the HTML tags is required. Grouping the texts and data into tags is required to easily group every text to be shown on the website. To be able to design the interface in terms of the positioning of texts and data, it must be linked to a CSS file. The next step after creating an HTML file is creating a CSS file. In coding in CSS, the tags from the HTML files will be used to generate and customize the layout of the website. Lastly, JavaScript coding will be used in addition to the HTML and CSS codes created.

3.6.1.2 Google Maps JavaScript API

The routing section located under the burger button of the website used4 Google Maps JavaScript API. This API abled to customize the maps on our website, pinning the three nodes on their respective locations, barangay hall, and the evacuation center for the user to have a clear image of where these locations are located within the barangay. An added feature of routing the user to the nearest passable road to the evacuation center will also be implemented.

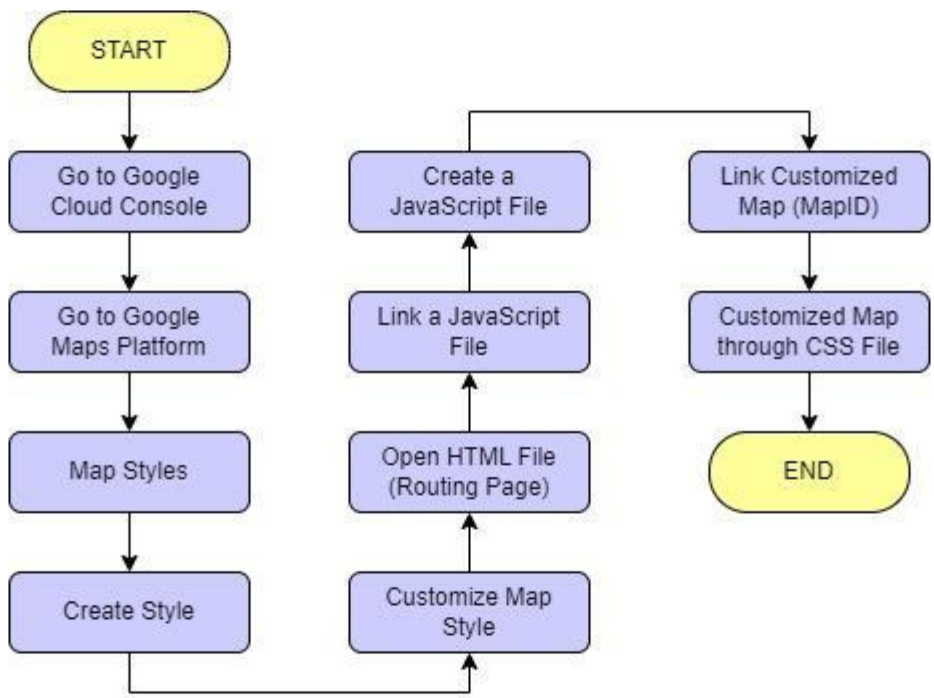


Figure 3.27 Process Flow for the Creation of Routing Page

Figure 3.27 shows the process flow for the implementation of a map on the routing page of the website. It begins by going to the Google Cloud Console online and then going to the Google Maps Platform. Select Map

Styles and then create a style; this will redirect to a page where maps can be customized. Save the customized map and its MapID to be used in the coding process. The next step is to open the created HTML file for the routing page and add a tag to link a JavaScript file. Create a Javascript file, and then link the customized map inside using the MapID. To position the map inside the website, CSS was used.

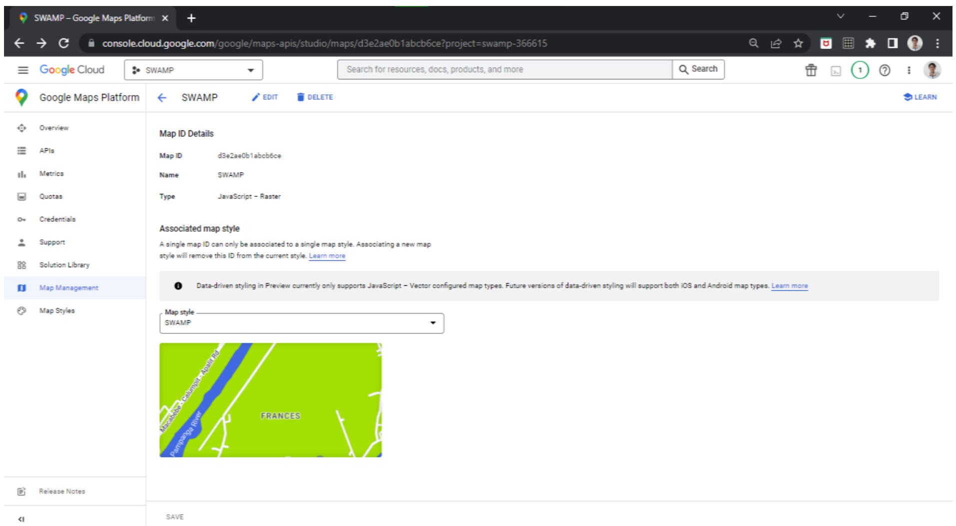


Figure 3.28 Google Maps JavaScript API

Figure 3.28 illustrates the integration and utilization of the Google Maps JavaScript API on the SWAMP website. This API was used to incorporate dynamic maps, geolocation, and interactive features into our website.

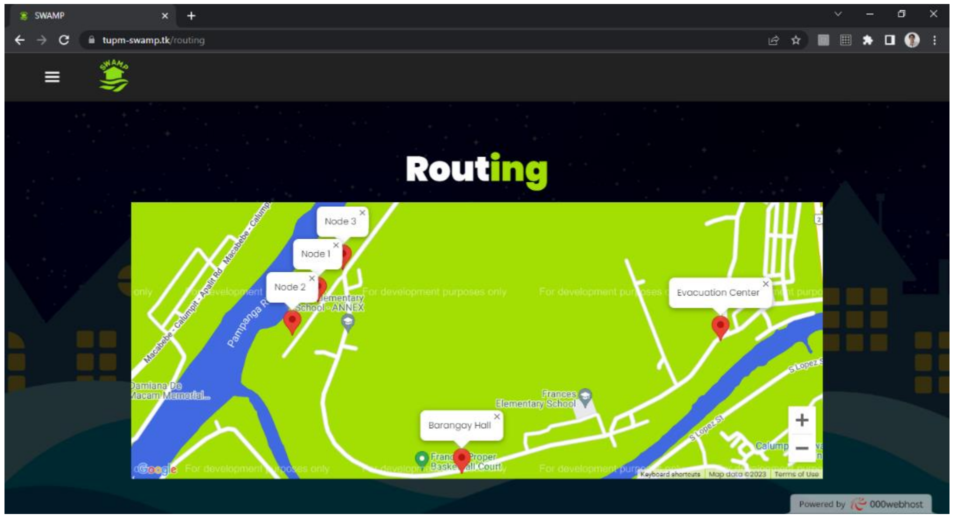


Figure 3.29 Routing Page Implementing the Google Maps JavaScript API

Figure 3.29 shows the Routing Page implementing the Google Maps JavaScript API. Users can interact with the map by zooming in/out, panning across different areas, and clicking on the pinned markers to access information about that specific location. It pins the locations of the three nodes, barangay hall, and the evacuation center of the Barangay Frances in Calumpit, Bulacan.

3.6.1.3 Web Server and Domain

The proponents tested and uploaded the website online on the Internet through 000webhost which is a free hosting website that uses HTML, CSS, and Javascript files. The website acquired its domain name system (DNS) from Freenom named tupm-swamp.tk. The DNS is free and will be available for 1 year with free to renew (unlimited renewals) provided by Freenom. In securing the hosted website online, Cloudflare was used. It will give protection to the purchased DNS to be not seen as a malicious website to any user.

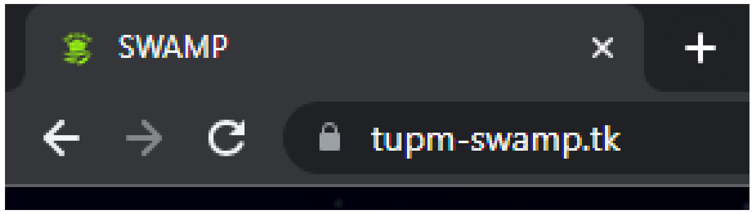


Figure 3.30 Website URL

Figure 3.30 is the url for our website. It can be accessed by typing "tupm-swamp.tk" on the web browser of laptops and mobile phones. It contains the name and logo of our website to be distinguished easily.

Domain	Registration Date	Expiry date	Status
tupm-swamp.tk	2022-11-21	2023-11-21	ACTIVE

Figure 3.31 Purchased DNS from Freenom

Figure 3.31 shows the purchased DNS (Domain Name System) of our website. It was obtained from Freenom that will be available for 1 year and is renewable free (unlimited).

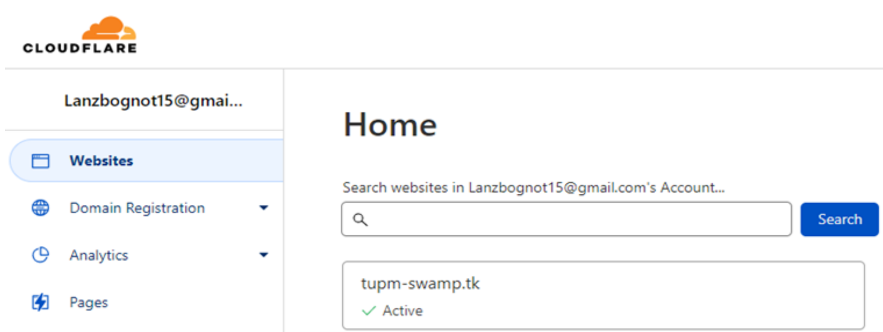


Figure 3.32 Securing the Website

Figure 3.32 shows the used cloud security platform of our website. Cloudflare was used to further help in securing our hosted website online. This helps in not detecting our website as a threat to the users' mobile devices and hence helps in securing it.

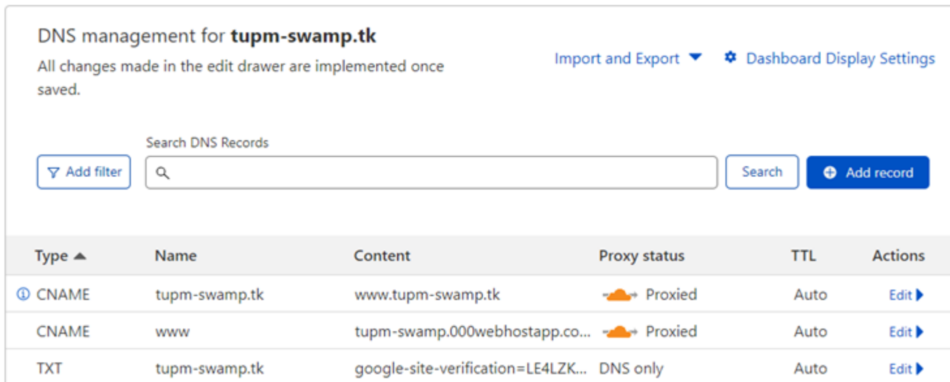


Figure 3.33 Linking of the Purchased DNS to Cloudflare

Figure 3.33 shows the linking of the purchased DNS from Freenom to Cloudflare. This will help protect the purchased DNS from any threats online.

3.6.2 Back-End development and database creation

3.6.2.1 XAMPP, Apache, MySQL, phpMyAdmin

In testing the created interface design, the developer used XAMPP which is a free and open-source cross-platform web server solution stack package that provides the Apache web server, MySQL database and php as command-line executable. This program will help the developer to create and test the created programs on a local web server. The Apache will be the web server that is responsible for accepting directory (HTTP) requests from

Internet users and would send them to a desired web page. MySQL was used as a database management system on the registration and login page of the website. This helped the proponents have a list of all the users that accessed the website. The phpMyAdmin served as the software that handled the administration of the MySQL database.

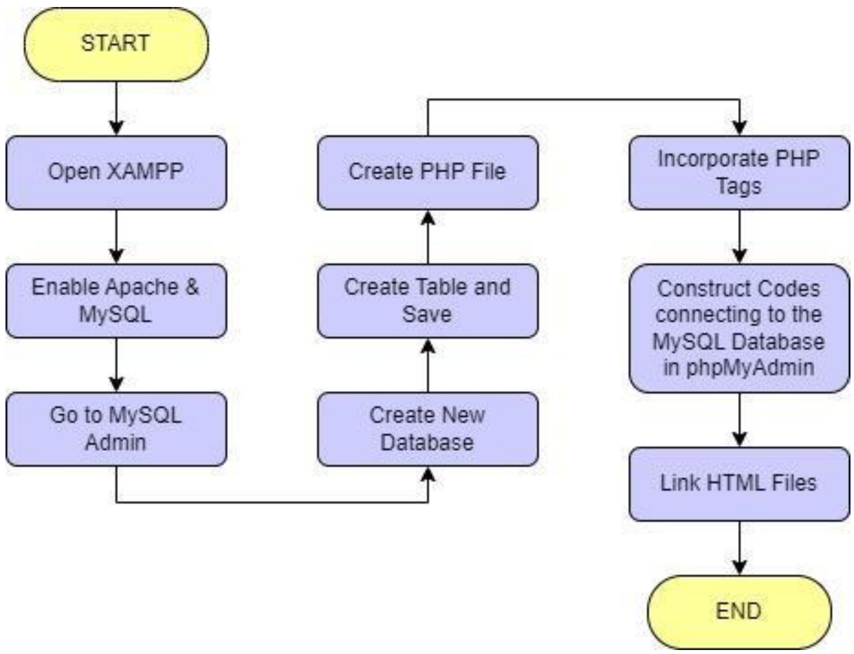


Figure 3.34 Process Flow for the Creation of Database for Testing

Figure 3.34 shows the process flow for the creation of a database using MySQL. To begin, open the XMAPP application that contains Apache and MySQL that will be used in the creation of the database. The XMAPP will serve as the web server for the website. The next step is to enable Apache and MySQL, and then go to the MySQL admin to be redirected to phpMyAdmin, where the database will be created. Create a new database, and then start to create and construct a table that will store the needed information from the website. After creating a database, a PHP file will be used to generate the created database. This PHP file needs to incorporate PHP tags to work. Construct codes relating to the designed interface and then connect it to the created MySQL database in phpMyAdmin. Linking the HTML files will also be considered for the redirection of the page when the saving of the information to the database is done.

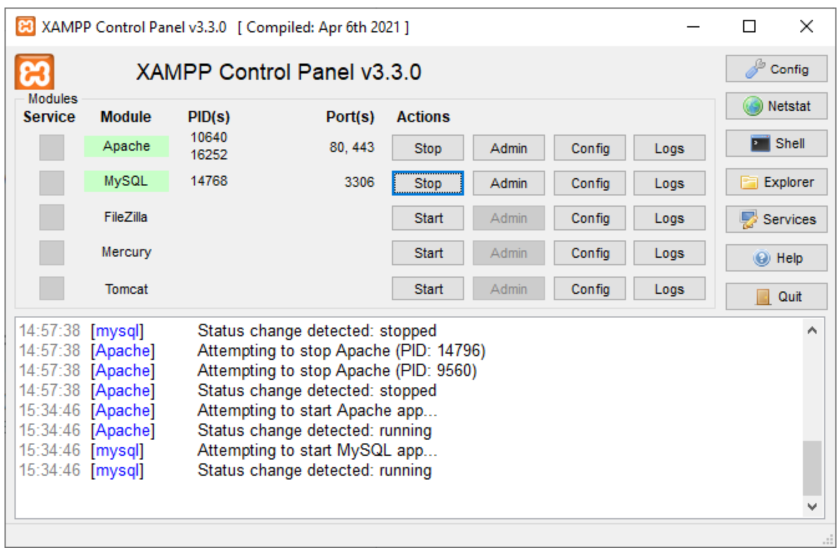


Figure 3.35 XAMPP Control Panel

Figure 3.35 displays the XAMPP Control Panel which is used in testing the created website on a local host to see how it would look when uploaded online on the webhost.

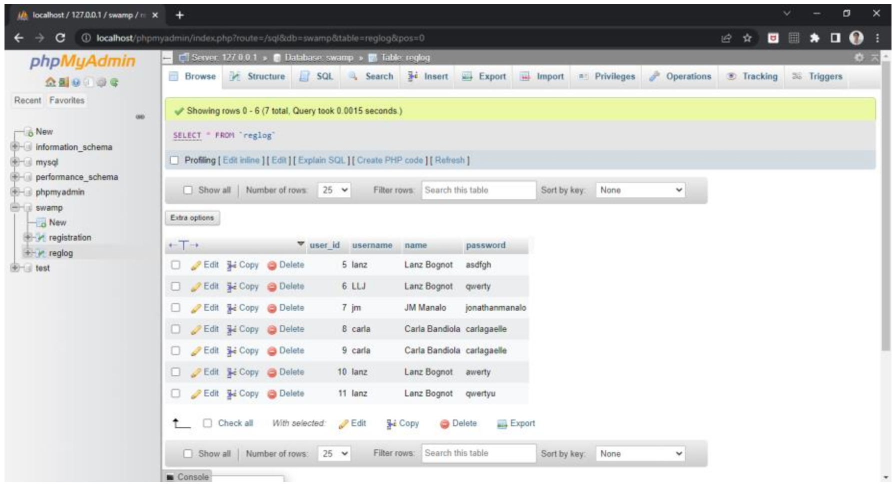


Figure 3.36 phpMyAdmin Database

Figure 3.36 shows the database that stores the credentials of the users when they accessed our website. This was made through using phpMyAdmin Database that is also available on the free webhost used on our website.

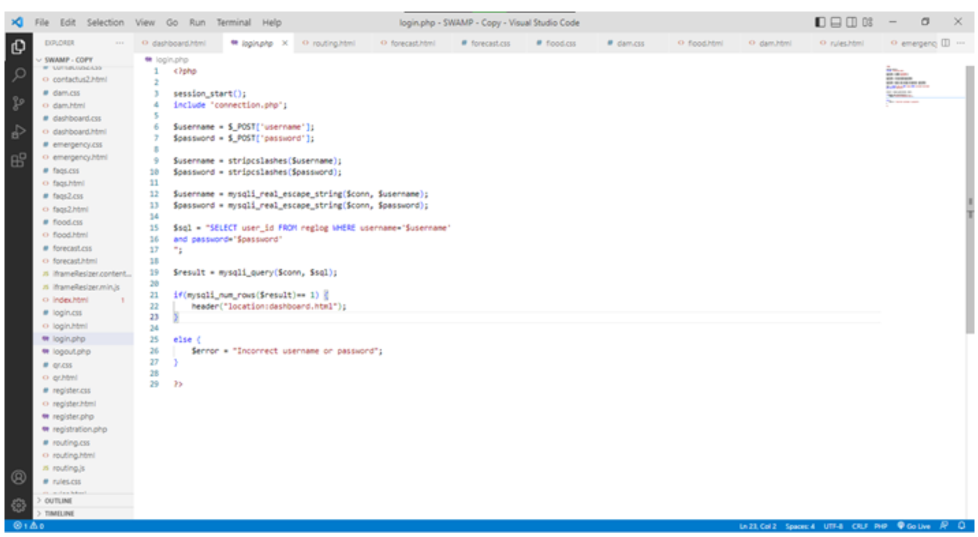


Figure 3.37 Sample php Codes

Figure 3.37 shows a sample of php codes used that is connected to the login and sign-up button of the website. When the button is clicked, these codes will run through and will then access the database created on phpMyAdmin.

3.6.3 Integrated Alert System

On behalf of the other residents who were not able to access the website due to the lack of available devices, an alert system will be implemented through a siren that will be connected to every node the proponents will build.

Figure 3.38 Calibration of Siren (ES-626)

The siren will act as an alert signal, only alarming when there is a rapid increase in water level and a flood is imminent. This new system will also be useful in informing those who are not that computer literate about how to use our website. Figure 3.6.3 shows the calibration and testing of the ES-626 siren.

3.6.4 Web-based Notification

The proponents will implement a web-based notification that uses a JavaScript Notification API to display desktop-style notifications that run in the browser using pure JavaScript. To make this work, the created website would need to have a secure context (https) for the developer to trigger the created codes to function on the users' end. The notifications will include an alert message covering the monitored and predicted data for the day, as well as notifications informing users that an increase in water level is occurring and that a flood may occur. If they granted and accepted notification from the website, this notification will appear on their devices automatically.

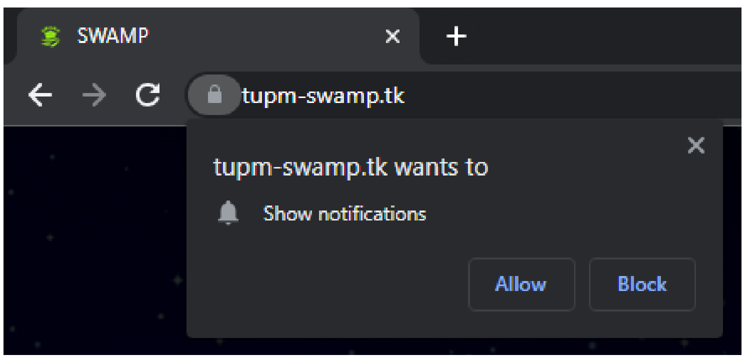


Figure 3.39 Sample Request Notification Permission

Figure 3.39 displays the request of notification permission to the user for our website. This notification when allowed by the user will then automatically send them updates regarding to flood occurrence. The users can also disable this by clicking block if they want to be not notified.

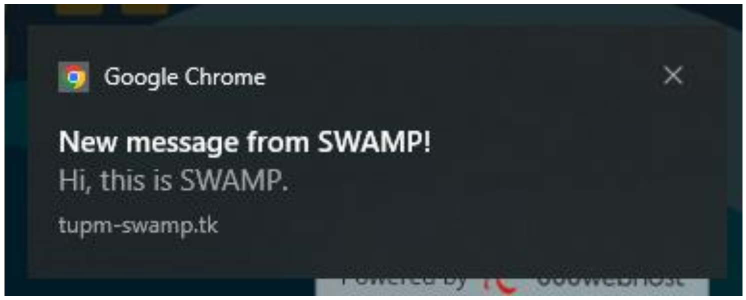


Figure 3.40 Desktop Notification

Figure 3.40 shows the desktop notification of the website. This will automatically show up when the user had agreed on the request of notification permission from the website. It contains information and data from the SWAMP website and when clicked the users will be redirected to the homepage of the SWAMP website.Figure 3.40 Sample Notification on Desktop

3.7 Testing and Evaluation of the Flood Monitoring, Prediction, and Alert Systems

The developed system and device will be tested to determine and evaluate the output status of the used sensors, including operability and accuracy of results. To determine the efficiency of the device and system, the MDRRMC of the Municipality of Calumpit, Bulacan together with the BDRRMC and residents of Barangay Frances in Calumpit, Bulacan will be surveyed.

3.7.1 Standards Used for Data Parameters

The tables below provide the standard measurements for each parameter needed in the study.

3.7.1.1 Flood Level

Table 3.4 Standard for Flood Level

Response	NDRRMC Standards (nches or ft)	Image Processing (ft)	Ultrasonic Sensor (cm)
Red Warning (Evacuate)	51 – 75 inches or 4.3 – 6.25 ft	4 – 6	
Orange Warning (Alert)	26 – 50 inches or 2.2 – 4.2 ft	2 – 4	
Yellow Warning (Monitor)	25 inches or less than 2.1 ft	1 – 2	

Barangay Frances and the local municipal provided the water level measurement standard for evacuation when floods occur. The barangay uses a ten (10) year water level measurement to refer to their evacuation process, from the Municipal of Calumpit, Bulacan.

3.7.1.2 Precipitation Rate

Table 3.5 Standard for Precipitation Rate

Response	Pagasa Rainfall Warning (mm/h)	Rain Gauge Sensor (in/h)
Red Warning (Evacuate)	More than 30 mm	
Orange Warning (Alert)	15 – 30 mm	
Yellow Warning (Monitor)	7.5 – 15 mm	

The standards of the precipitation rate came from the color-coded rainfall advisories of PAGASA from the Official Gazette. The data from the rain gauge will be converted to millimeter per hour to fit the measurements of the PAGASA rainfall advisory.

3.7.1.3 Atmospheric Pressure

Table 3.6 Standard for Atmospheric Pressure

Classification	International Standard	BMP 180 Sensor (hPa)
High	1083.8 mbar	
Normal	871 – 1083 mbar	
Low	870 mbar	

The atmospheric standard follows an international standard measured from sea-level pressure. This standard measurement follows the common barometric pressure done at weather reports. The BMP 180 Sensor data conversion to millibar will be the same as each value equates to itself.

3.7.1.4 Dam Water Level and Gate Opening

Table 3.7 Standard for Dam Information

DAM	Normal High Water Level (NHWL) (m)	Reservoir Water Level (RWL) (m)	Gates Opening	
			Gates	Meters
Angat	210			
Ipo	101			
Pantabangan	221			

The standard measurements of the dam information is also from the website of PAGASA. The website displays various parameters of the dam water level and spillage – the table above uses the parameters fit for the project study. Each parameters have uniform measurements which eliminates the conversion of data.

3.7.1.5 Siren Sound for Evacuation

Signal	Meaning	Action	Siren Sound
Wavering Tone for Three (3) Minutes	Flooding Occurs	Evacuation	500 Hz to 1500 Hz
Steady Tone for Three (3) Minutes	Flooding has Subsided	Return to Respective Houses or Normal Activities	500 Hz to 1500 Hz
Steady Tone for Thirty (30) Seconds	Testing Sound	Every Tuesday at 1 PM	500 Hz to 1500 Hz

Table 3.8 Standard for Siren Sound for Evacuation

The standard siren sound is from the Davao City Risk Reduction Management Office Webpage. The webpage provided a YouTube video that showcased the three different signals for disaster alert signals. The meaning and action of the standard was reduced to fit the flood warning and evacuation process of the project. The siren will use the standard frequency for each signal that will be used by the actual siren component.

3.7.2 Testing Procedure

Tests will be conducted to determine the accuracy of each sensor reading in the flood monitoring, prediction, and alert systems.

Table 3.9 Flood Detection, Precipitation Rate, and Atmospheric Pressure Readings

Date	Time	Flood Level (cm)	Flood Level (ft)	Precipitation Rate (in/h)	Atmospheric Pressure (hPa)	Dam (m)

Table 3.9 shows the flood level from the ultrasonic sensor (centimeters), flood level from the web-camera module (feet), precipitation rate (inches per hour), atmospheric pressure (hectopascals), and dam (meters) readings, as well as the date and time they were taken.

3.7.2.1 Accuracy Testing and Validation of Predictive Algorithm

The Accuracy Testing and Validation of the Predictive Algorithm uses Mean Absolute Error as the determinant of the program's performance. It compares the actual and predicted flood levels for one week worth of hourly data in determining its Mean Absolute Percentage Error with respect to date and time.

Table 3.10 Parameters for Accuracy Testing and Validation

Date and Time	Actual Flood Level	Predicted Flood Level	Mean Absolute Percentage Error

The output values for predicted flood levels are then compared and computed with the actual flood levels. The formula of Mean Absolute Percentage Error and Mean is used to determine the performance of the model in each instance and its average.

$$MAPE = \frac{Forecasted\, Value - Actual\, Value}{Actual\, Value}$$

$$Mean\, Value = \frac{Sum\ of\ Values}{Number\ of\ Values}$$

The output values will be used for the decision classifier. The decision classifier uses the standards given by PAGASA mentioned in Table 3.4 and Table 3.5.

3.7.2.2 Accuracy Testing and Validation of Image Processing

The Accuracy Testing and Validation for Image Processing uses Confusion Matrix in determining the model’s performance. It compares the true positives and true negatives, as well as its false counterparts to identify how well the program performs. The program uses 600 images extracted from the frames of an actual video taken from Angat River. The performance is validated by actual object detection.

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Figure 3.41 Confusion Matrix

3.7.3 Evaluation Procedure

The flood monitoring system will be evaluated by the Municipal Disaster Risk Reduction and Management Council (MDRRMC) of the Municipality of Calumpit, Bulacan, including the Barangay Disaster Risk Reduction and Management Council (BDRRMC) and residents of Barangay Frances in Calumpit, Bulacan. They will complete a technical evaluation form based on ISO 9126-1, the international standard for software evaluation relating to the device system's functionality, reliability, usability, efficiency, maintainability, and portability. The evaluation form aims to determine whether the device is user-friendly, cost-effective, dependable, and conventional.

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: ______________________________ Petsa: ____________
Address: ______________________________ Kasarian: __________ Edad: _______
Contact Number: ______________________ E-mail: ______________________

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

		Higit na hindi sumasang-ayon	Hindi sumasa ng-ayon	Sumasa ng-ayon	Higit na sumasa ng-ayon
	Katangian	**1**	**2**	**3**	**4**
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*				
2.	Will it not cause inconvenience upon use? *(Ito ba hindi ay nag-dudulot ng abala?)*				
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*				
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*				
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*				
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*				
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*				
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*				
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*				
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*				
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*				
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*				
2.	Is the authentication of users is easily processed?				

Transaction ID	
Signature	

Figure 3.42 Evaluation Form

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*				
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*				
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*				
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*				
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*				
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				

Katanungan o Komento:

SIGNATURE (LAGDA):

Transaction ID	
Signature	

Figure 3.43 Evaluation Form

Figure 3.42 to 3.43 shows a sample of an evaluation form to be answered by selected flood monitoring and disaster management experts, as well as barangay officials and residents of Barangay Frances in Calumpit, Bulacan.

3.8 Statistical Analysis

The Likert scale would be used for the statistical analysis in calculating the average mean of the user's responses on the evaluation form. A Likert-type scale consists of a series of statements from which respondents can select to rate their responses to evaluative questions. Table 2 shows the grading system that will be used on a 4-point Likert scale.

where:

f_i = number of respondents

i = Likert Scale Scores, SD (1), D (2), A (3), SA (4)

Formula for the determination of Likert scale total scores:

$$\textit{Total Scores} = \sum (f_i \times i)$$

Formula for the determination of Likert scale mean scores:

$$\textit{Mean Score} = \sum (f_i \times i) / \textit{Number of Respondents}$$

Table 3.11 Grading system

Description	Rating
Strongly Agree	3.5 – 4.00
Agree	2.5 – 3.24
Disagree	1.75 – 2.4
Strongly Disagree	1.00 – 1.74

3.9 Project Work Plan

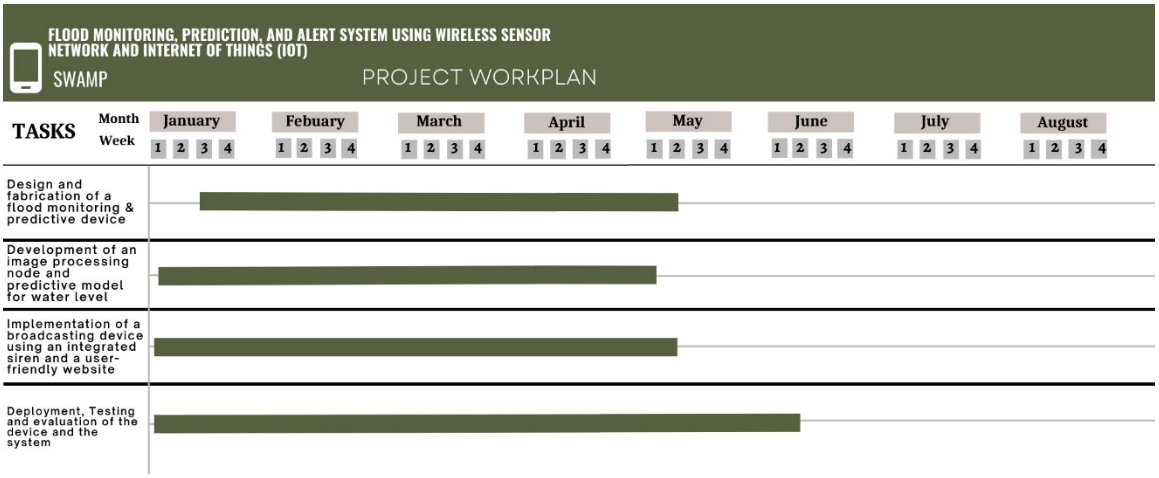


Figure 3.44 Gannt Chart for SWAMP

CHAPTER 4

RESULTS AND DISCUSSIONS

Results and discussion is an essential element of research studies, offering a platform to present and analyze findings. This section examines the findings in light of prior studies, draws relevant inferences, and provides insight into the study's broader implications.

4.1 Project Technical Description

The SWAMP system consists of three main components: monitoring, prediction, and notification. To monitor the water level, an ultrasonic sensor is used, while the barometric pressure sensor monitors the atmospheric pressure in the area. The image processing system utilizes a web digital camera to capture the water level gauge, providing real-time flood level information. Additionally, a time series autoregression model utilizes these parameters, excluding the image processing output, to ensure accurate flood level predictions. The system's website displays the sensor readings, camera images, and flood level predictions, while an integrated siren amplifies to alert residents in the event of a flood.

The entire system is powered by a Raspberry Pi 4B microcomputer. Data collection from various sensors, including ultrasonic, barometric pressure, and rain gauge, is exclusively done by the NodeMCU V3 microcontroller. The collected data is then transmitted to the ThingSpeak Cloud DataStream platform, which is connected to a website for display, analysis, and prediction purposes. Real-time flood-level viewing is facilitated by a web-camera connected to the microcomputer, which also includes integrated image processing capabilities. For flood level forecasting, the system utilizes a Time Series - Long Short-Term Memory (LSTM) prediction model, providing predictions for the next 3 hours. The website further features a routing system that showcases the locations of the three nodes and the community's evacuation center. Additionally, the website includes disaster risk management information, emergency hotlines, and an alerting mechanism. This alerting aspect involves website notifications and an integrated siren within the system. The system's power supply comprises an 80 Watts solar panel, a 12V solar charge controller, a 12V or 25Ah lead-acid battery, and a DC to AC inverter, ensuring reliable and sustainable operation.

4.2 Project Structure

4.2.1 Overall Project Structure

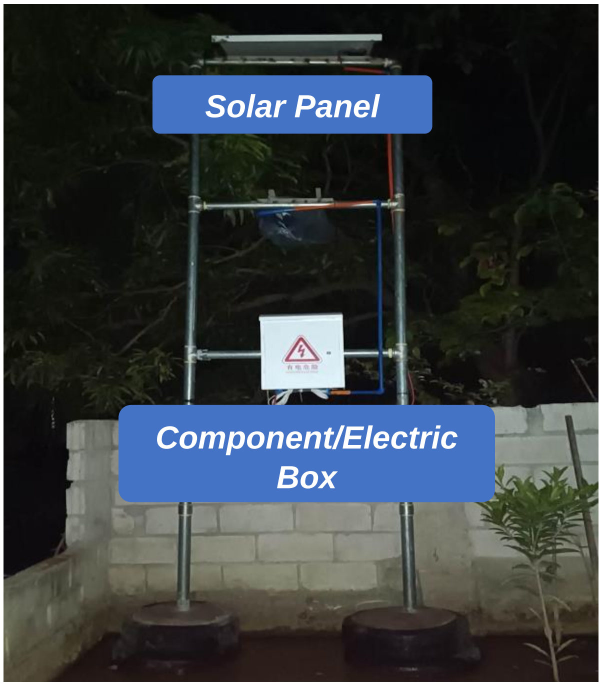


Figure 4.1 Overall Project Structure

Figure 4.1 shows the SWAMP setup at the deployment site while providing a modular assembly. The highest pole contains the solar panel, the second row of poles holds the platform for the integrated sensors at three (3) meters high, and the first row includes the panel box. The overall height of the pole is four (4) meters, and its connecting poles are one (1) meter.

4.2.2 Integrated Sensors

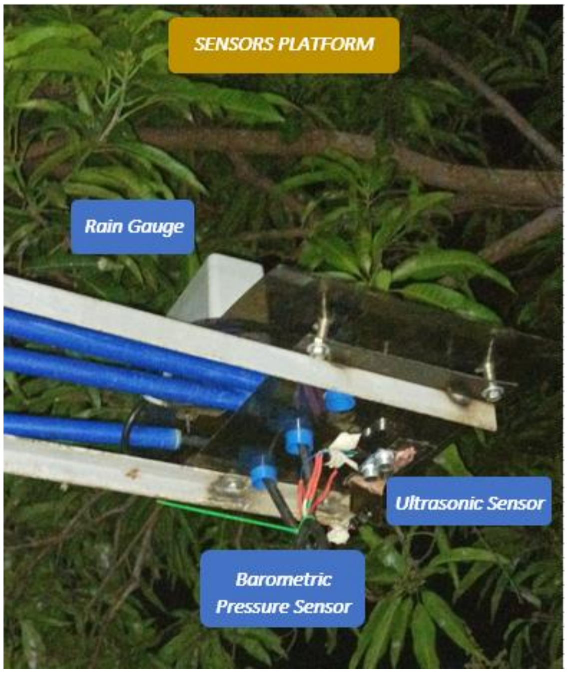


Figure 4.2 Integrated Sensors

Figure 4.2 shows the integrated sensors with the platform used for the flood monitoring and prediction system. The rain gauge, barometric pressure sensor, and ultrasonic sensor connect to the NodeMCU and sends it to the cloud data stream.

4.2.3 Microcomputer, Microcontroller, and Power Supply

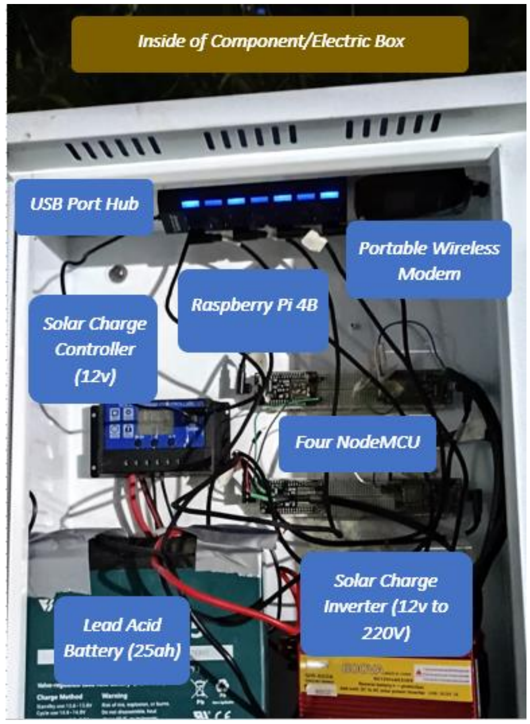


Figure 4.3 Microcomputer, Microcontroller, and Power Supply

Figure 4.3 shows the collection of the microcomputer, microcontroller, power supply, and their connection inside the panel box. This section provides the backend program and the communication of the system.

4.2.4 Image Processing System

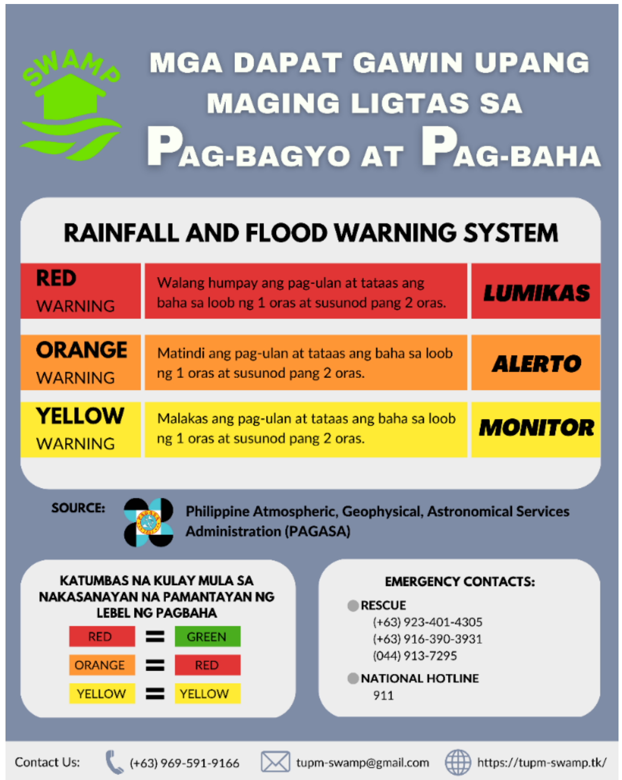


Figure 4.4 Flood Warning System Standard

Figure 4.4 shows the flood warning standard used in the project by Philippine Atmospheric, Geophysical, and Astronomical Services Administration (PAGASA).

Figure 4.5 Web Digital Camera

Figure 4.5 shows the web digital camera for image processing. It provides a real-time display of the water level from the water level gauge.

Figure 4.6 Water Level Gauge

Figure 4.6 shows the water level gauge used for image processing. The design of the gauge is also in modular form.

4.3 Project Limitations and Capabilities

The project focuses on monitoring, predicting, alerting, and notifying about ongoing floods through the utilization of a wireless sensor network and the Internet of Things (IoT). The prediction model incorporated in the system can forecast the flood level with a three-hour lead time before it occurs. Additionally, the system features geographic mapping and routing functionalities. It is important to note that while the system can predict the flood's height, it does not provide information regarding the duration of the flood.

4.4 Project Testing

4.4.1 Graph of Collected Data from the Sensors

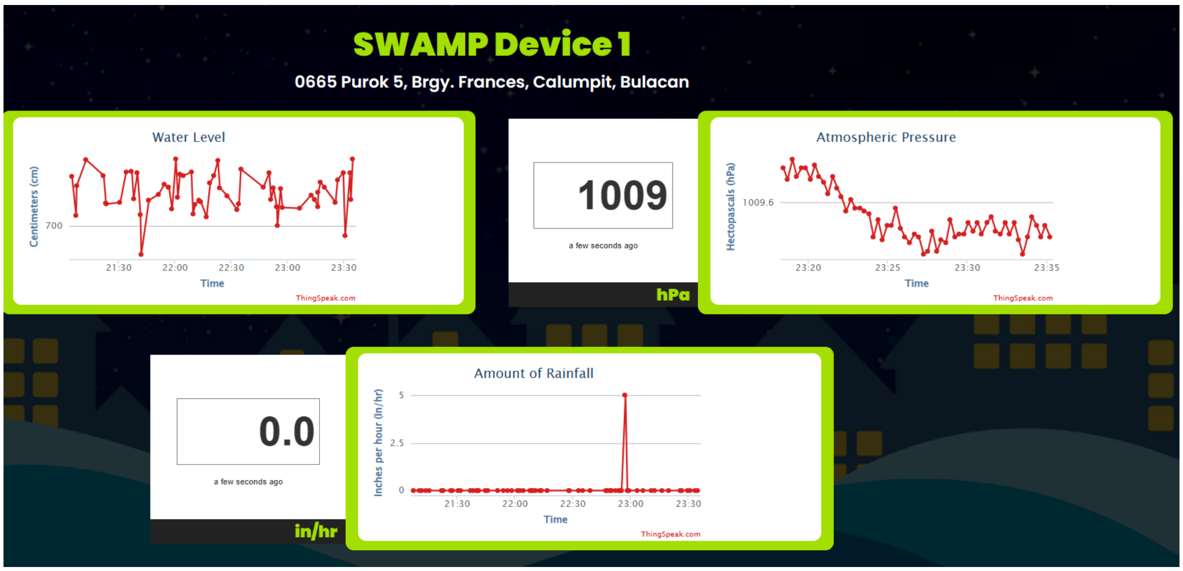


Figure 4.7 Real-Time Data Gathering of Hydrological Sensors from Node 1

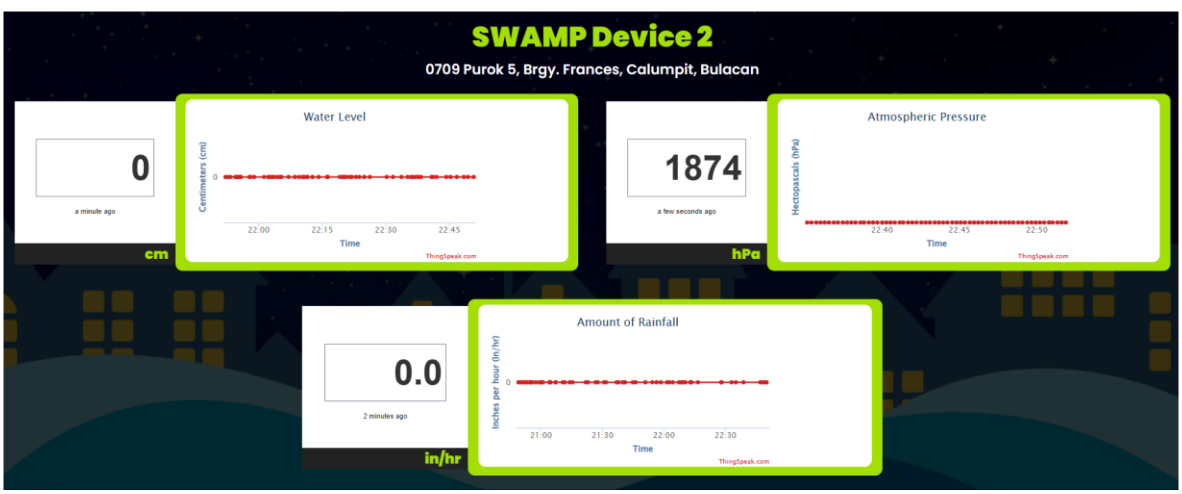


Figure 4.8 Real-Time Data Gathering of Hydrological Sensors from Node 2

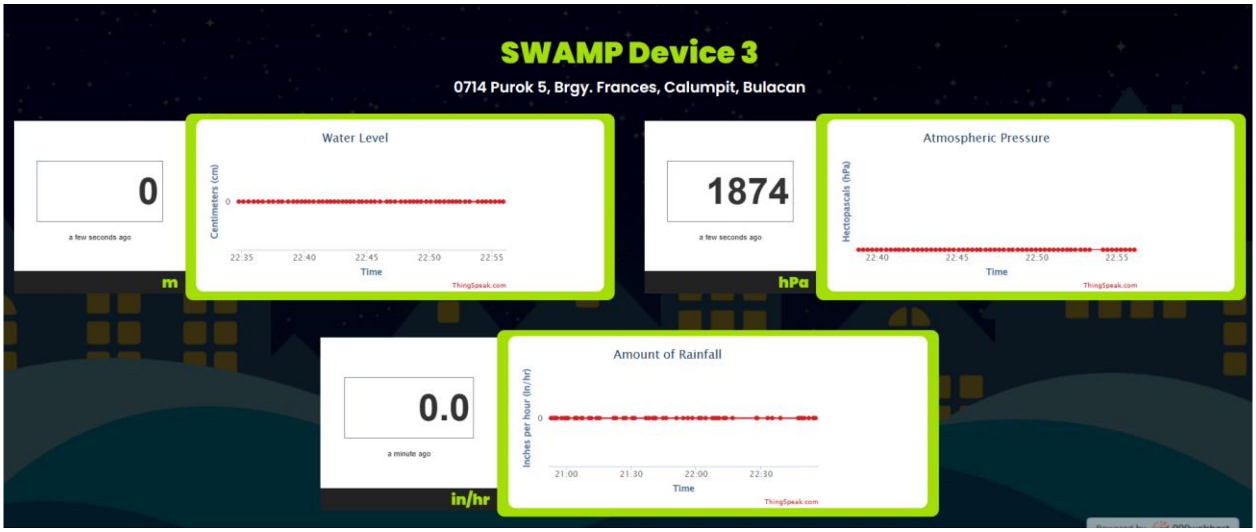


Figure 4.9 Real-Time Data Gathering of Hydrological Sensors from Node 3

The hydrological sensors collect data, which is transmitted to the ThingSpeak platform for real-time monitoring of water level, atmospheric pressure, and rainfall amount. The ultrasonic sensor provides accurate measurements but may have outliers due to external factors, so a median filter was implemented. It collects twenty readings, discards the five highest and lowest values, averages the remaining readings, and displays the result on the website. This helps reduce outliers in the dataset. The barometric pressure sensor provides precise measurements that correspond with the standard atmospheric pressure level from PAGASA. The rain gauge accurately measures rainfall by converting the number of tips into the amount of rainfall. To improve accuracy and reliability, the reed switch in the rain gauge was replaced with a hall effect sensor.

4.4.2 Results of Image Processing

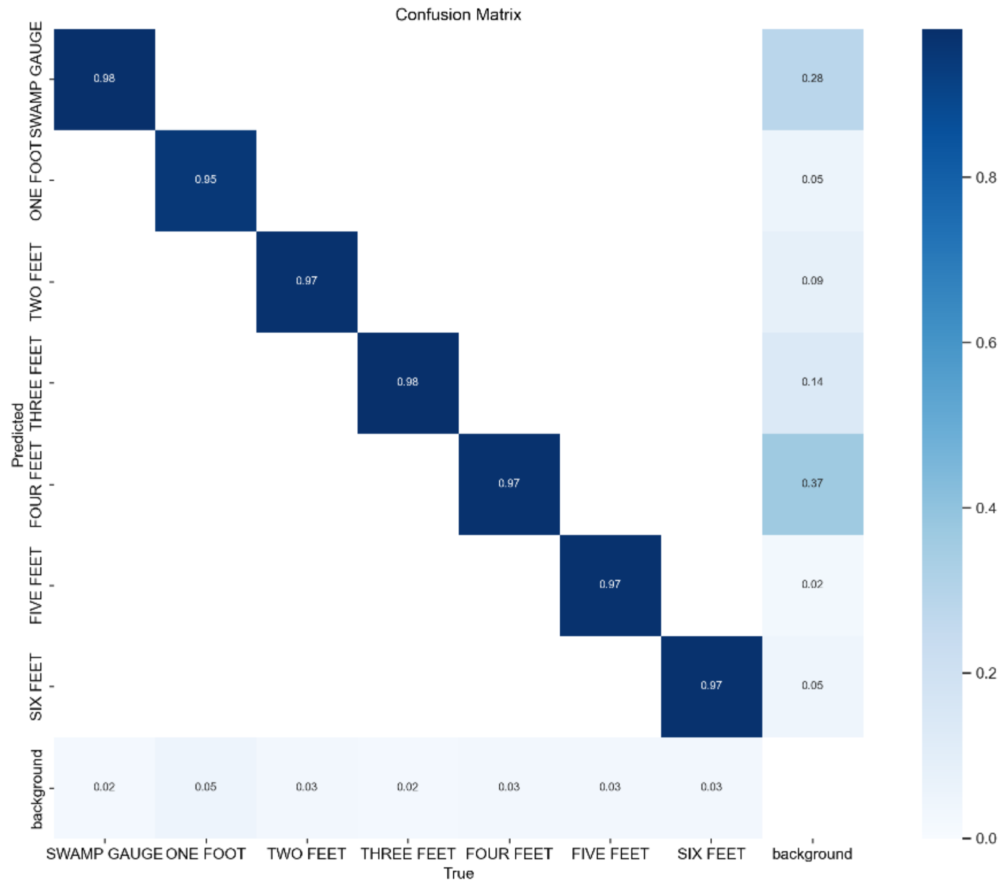


Figure 4.10 Confusion Matrix for the Trained Dataset

Figure 4.10 shows the confusion matrix of YOLOv5. The graph represents the accuracy of each class from the trained dataset. The darker colors represent higher accuracy, while the lighter colors represent lesser accuracy of the program.

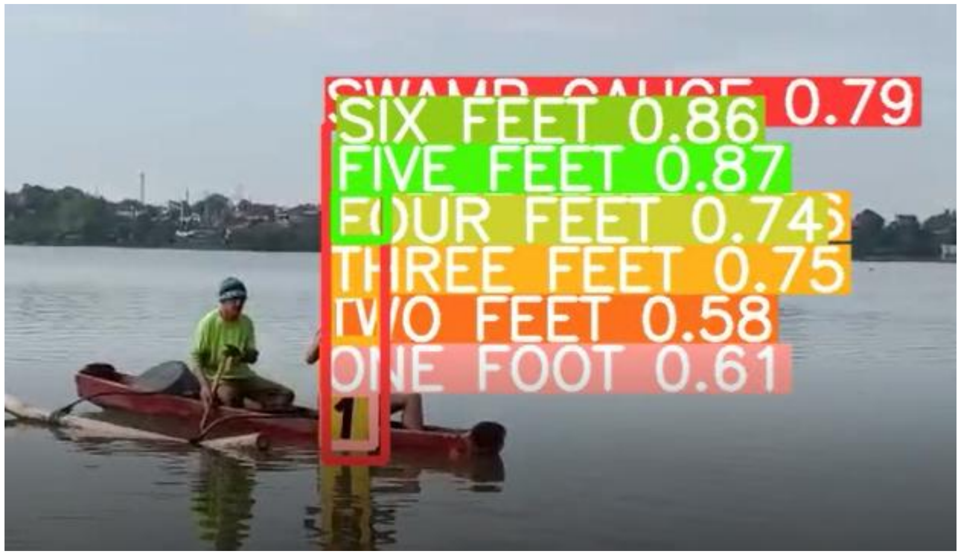


Figure 4.11 Results of the Test Video with Increased Water Level

Figure 4.11 above displays result of the training video with actual increase of water level. The video showed higher confidence level from each class as the initial trained dataset were gathered there.

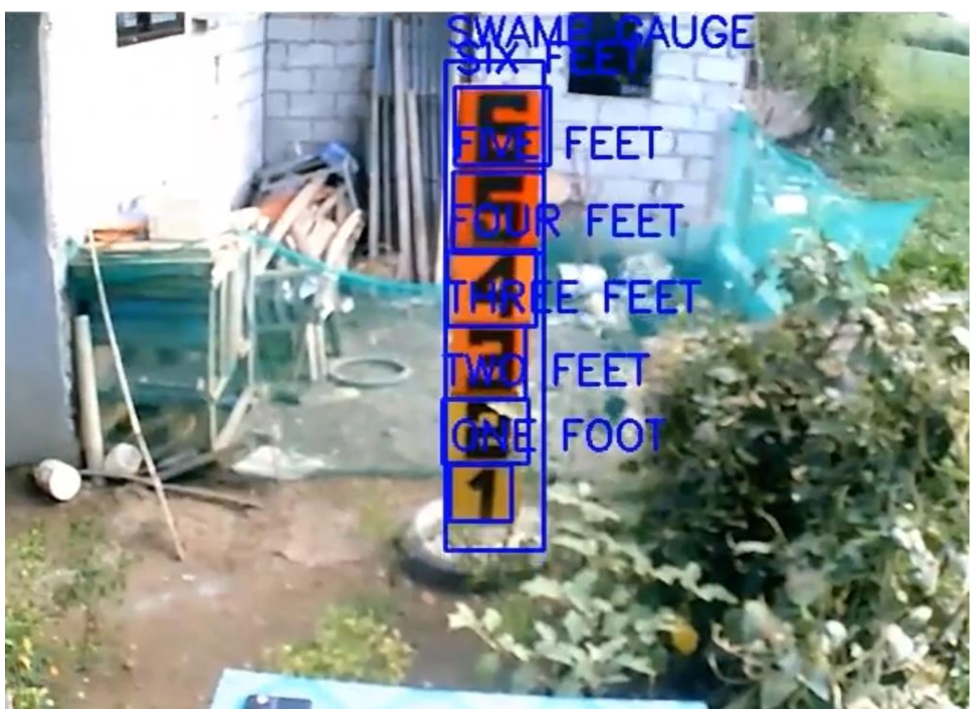


Figure 4.12 Test Results of Image Processing from the Water Level Gauge

Figure 4.12 executes the image processing from the actual water level gauge used in the system. Each number equates to a class with feet as their unit and an additional class called the swamp gauge. The webcam livestreams the feed to the website with the embedded YOLOv5 model.

4.4.3 Results of Flood Level Prediction

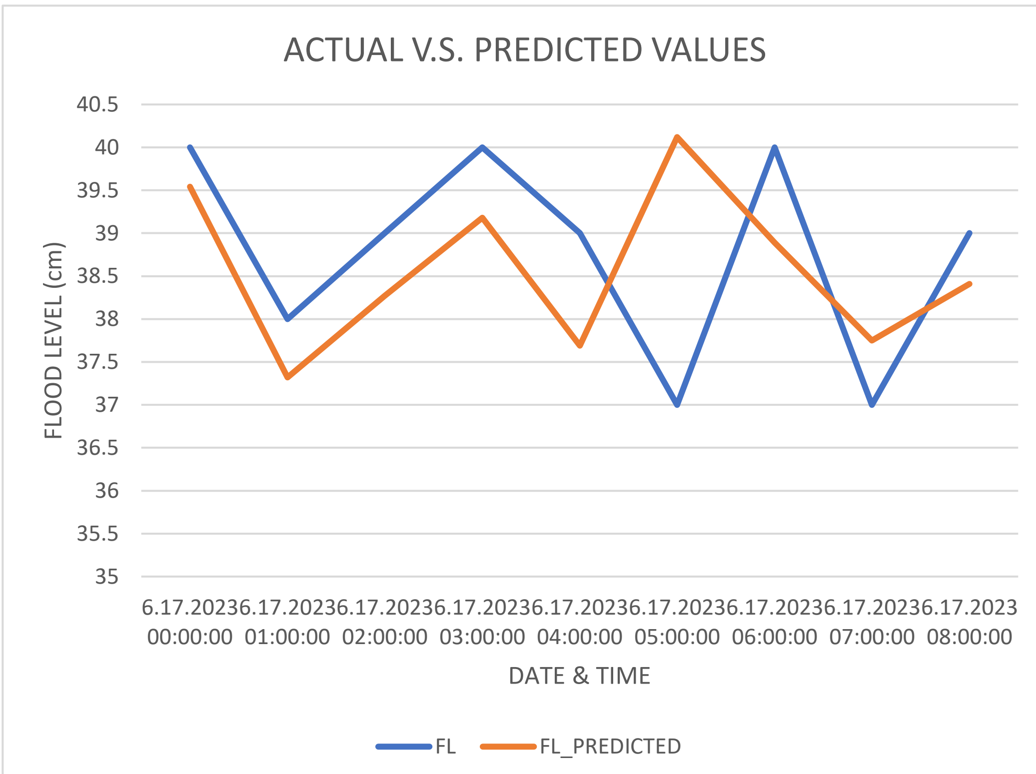


Figure 4.13 Graph of Flood Level Prediction

The provided information presents a comparison between the projected flood level (Predicted Flood Level) and the observed flood level (Actual Flood Level). The accuracy of the projections is evaluated using metrics such as the mean absolute percentage error (MAPE). The MAPE is a commonly used measure to assess the accuracy of forecasts, quantifying the level of accuracy achieved.

$$MAPE = \frac{Forecasted\,Value - Actual\,Value}{Actual\,Value}$$

The analysis of data reveals that the predicted flood levels and the actual flood levels are relatively close to each other. The Mean Absolute Percentage Error (MAPE) values, which indicate the average percentage difference between the projected and actual values, range from 0.05% to 10.59%. These values suggest that the predicted flood levels are reasonably accurate.

In terms of performance, both the MAPE and Accuracy metrics indicate the significance of the prediction model in forecasting flood levels. The high Accuracy values signify a strong agreement between the anticipated and real values. Additionally, the lower MAPE values indicate a comparatively low average percentage variance between the expected and actual flood levels. This suggests that the prediction model can provide valuable insights for predicting flood levels.

Table 4.1 Tabulated Averaged Data of Actual Flood Level and Predicted Flood Level

AVERAGE VALUES		
Actual Flood Level	**Predicted Flood Level**	**Average Percentage Error**
38.54	39.00	1.18%

The average flood level (Actual Flood Level) in the dataset is estimated to be 38.54. This value represents the mean of all the recorded flood level measurements, giving an indication of the typical or average value observed.

Similarly, the average predicted flood level is 39.00. This value is derived from the average of all the forecasted flood level numbers generated by the forecasting model. It serves as an approximate representation of the average or central tendency of the predicted flood levels.

$$Mean\ Value = \frac{Sum\ of\ Values}{Number\ of\ Values}$$

$$MAPE = \frac{Average\ Forecasted\ Value - Average\ Actual\ Value}{Average\ Actual\ Value}$$

The forecasting model for predicting flood levels demonstrates a high degree of accuracy, with the average percentage variation being just 1.18%. This minimal average deviation indicates that the anticipated flood levels closely align with the actual flood levels, validating the model's effectiveness.

In summary, the low average deviation and high average accuracy numbers emphasize the forecasting model's capability in accurately predicting flood levels. These metrics serve as reliable indicators of the model's accuracy and reliability in flood level predictions.

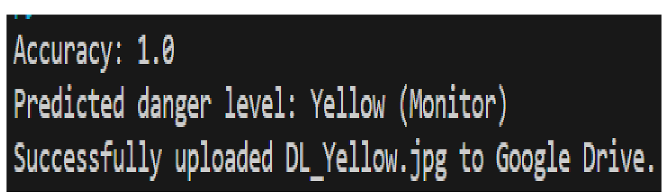


Figure 4.14 Prediction of Flood Level

Figure 4.15 Prediction of Flood Level

The figures above shows the program output for the Decision Classifier. The Decision Classifier uses the output data of the prediction model and identifies the danger level of the predicted flood level. The Classifier then uploads its decisions to Google Drive where it is extracted and used for the web application.

4.4.4 SWAMP Website

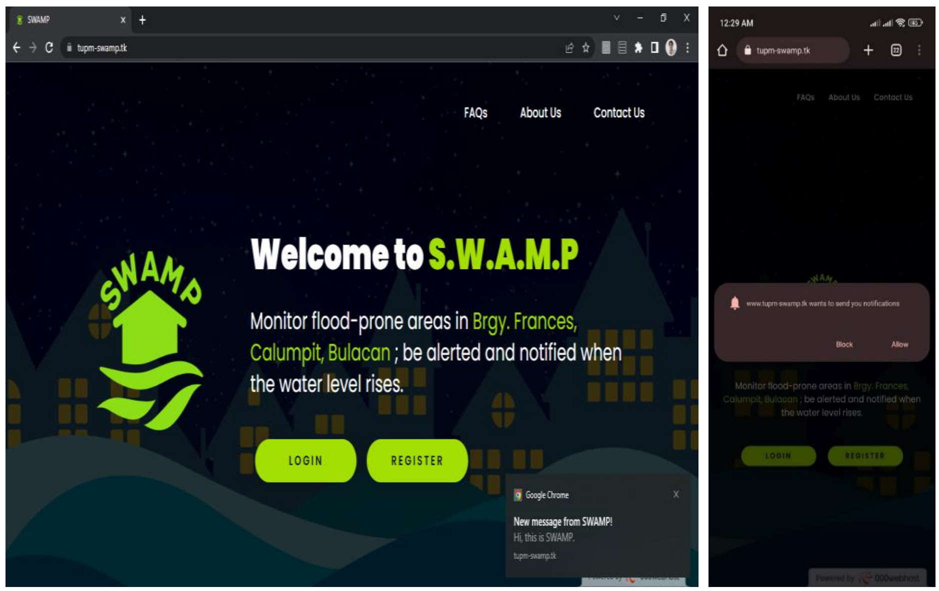


Figure 4.16 tup-swamp.tk

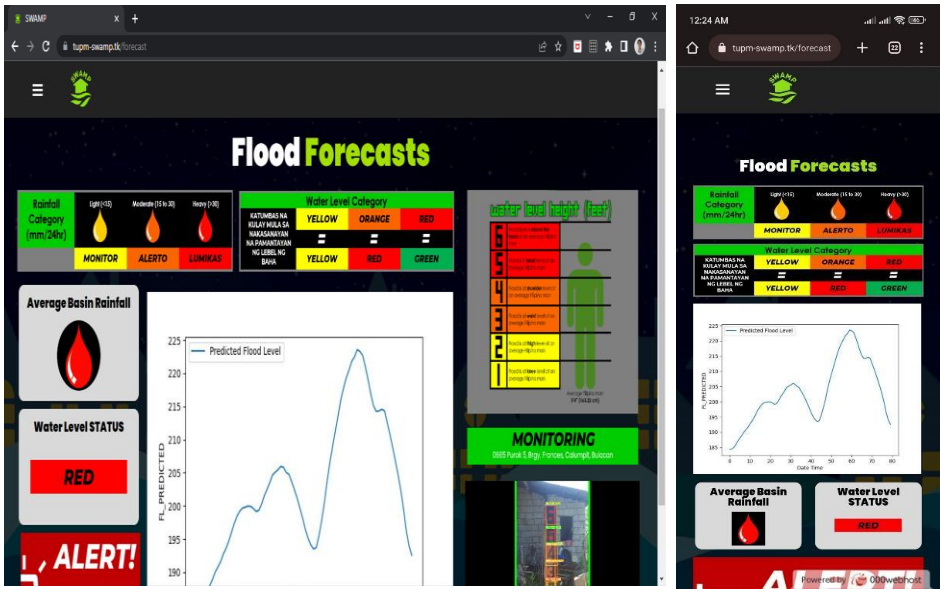


Figure 4.17 Flood Forecasts (Prediction, Monitoring, Alert)

The website "tupm-swamp.tk" provides a range of features to assist users in flood preparedness. Accessible via both laptops and mobile phones, the website offers a prediction graph and displays a rainfall and water level classifier using three colors (yellow, orange, red). The predictive model is stored on the Google Cloud Console. Real-time sensor data, including numerical values and graphs, is seamlessly integrated into the website. Secure sign-up and log-in functionalities ensure user privacy and convenience. The website provides comprehensive information to help users make informed decisions and effectively prepare for potential flooding incidents. With its dedicated server and user-friendly interface, it serves as a valuable resource for timely and reliable flood-related information.

4.5 Project Evaluation

4.5.1 Project Presentation in Barangay Frances

Figure 4.18 Project Presentation with the Sangguniang Barangay

Figure 4.19 Project Presentation with the Sangguniang Barangay

Figure 4.18 and 4.19 exhibits the project presentation of the researchers with their thesis adviser. The group discussed the history, significance, process, and output of the deploying system in their barangay. The adviser reviewed the

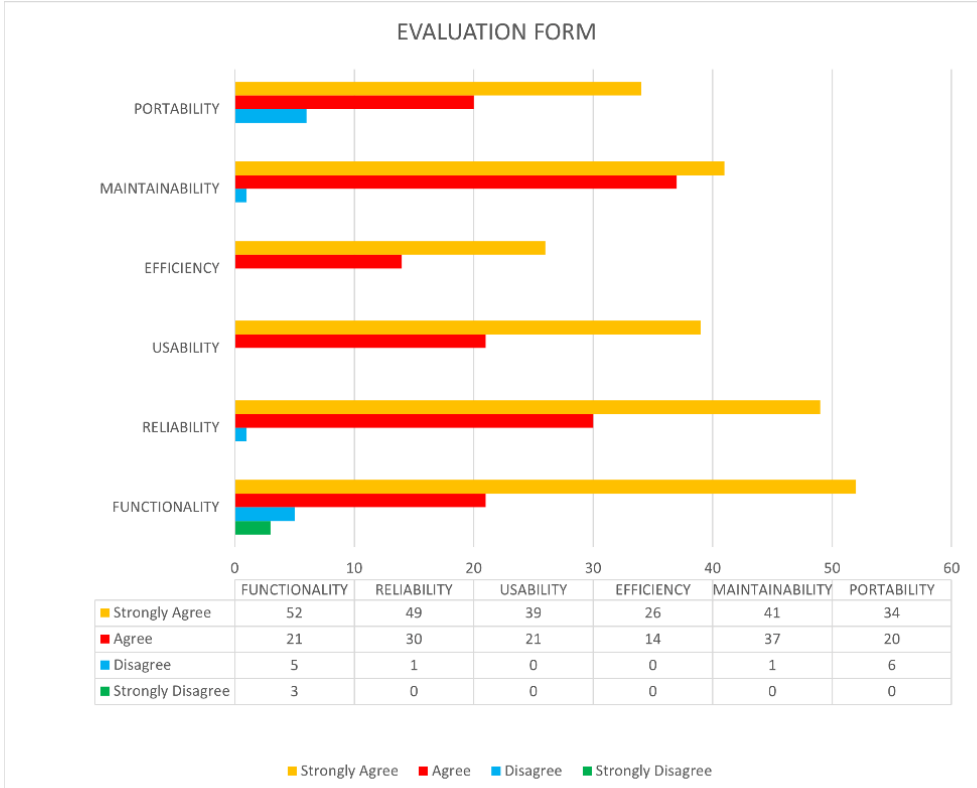
Memorandum of Agreement (MOA) and Memorandum of Understanding (MOU) with the barangay officials.

4.5.2 Site Visit of the Deployed Project

Figure 4.20 Site Visit of the Deployed Project

Figure 4.20 exhibits the deployed project presentation of the researchers with their thesis adviser to the panelist. The group discussed the obtained objectives and showed the working system from the power supply, integrated sensors, image processing, and predictive model program. The hardware aspect also includes the modular poles, mounting of the sensors, panel box, and the water level gauge.

4.5.3 Evaluation of SWAMP Device



	FUNCTIONALITY	RELIABILITY	USABILITY	EFFICIENCY	MAINTAINABILITY	PORTABILITY
Strongly Agree	52	49	39	26	41	34
Agree	21	30	21	14	37	20
Disagree	5	1	0	0	1	6
Strongly Disagree	3	0	0	0	0	0

Figure 4.21 Evaluation of SWAMP Device

The evaluation form received responses from 20 respondents in Barangay Frances regarding the SWAMP device. In terms of functionality, 52 respondents strongly agreed, 21 agreed, 5 disagreed, and 3 strongly disagreed. For reliability, 49 respondents strongly agreed, 30 agreed, 1 disagreed, and none strongly disagreed. In the usability category, 39 respondents strongly agreed, 21 agreed, and none disagreed or strongly disagreed. Regarding efficiency, 26 respondents strongly agreed, 14 agreed, and none disagreed or strongly disagreed. In terms of maintainability, 41 respondents strongly agreed, 37 agreed, 1 disagreed, and none strongly disagreed. Lastly, for portability, 34 respondents strongly agreed, 20 agreed, 6 disagreed, and none strongly disagreed.

4.5.4 Implementation of a 4-point Likert Scale to the Evaluation Results

Table 4.2 Implementation of a 4-point Likert Scale to the Evaluation Results

Statements	Mean Score	Description	Interpretation
Functionality	3.51	Strongly Agree	The respondents strongly agree that SWAMP exhibits high functionality, effectively fulfilling its intended purpose.
Reliability	3.6	Strongly Agree	The respondents strongly agree that SWAMP demonstrates exceptional reliability, consistently providing accurate flood monitoring, prediction, and alert services.
Usability	3.65	Strongly Agree	The respondents strongly agree that SWAMP demonstrates exceptional reliability, consistently providing accurate flood monitoring, prediction, and alert services.
Efficiency	3.65	Strongly Agree	The respondents strongly agree that SWAMP showcases remarkable efficiency, efficiently processing and analyzing flood data to deliver timely and relevant information.
Maintainability	3.46	Strongly Agree	The respondents strongly agree that SWAMP exhibits excellent maintainability, allowing for easy upkeep and ensuring the system's long-term functionality.
Portability	3.47	Strongly Agree	The respondents strongly agree that SWAMP displays exceptional portability, enabling seamless access and utilization across various devices and platforms.
Overall Mean Score	3.56	Strongly Agree	

1.0-1.74 (Strongly Disagree), 1.75-2.4 (Disagree), 2.5-3.24 (Agree), 3.25-4.00 (Strongly Agree)

Table 4.2 presents the evaluation of the SWAMP device using a 4-point Likert scale. The assessment included statements regarding functionality, reliability, usability, efficiency, maintainability, and portability. Respondents provided ratings ranging from 1 to 4, and percentages and total scores were calculated for each statement (refer to the appendix). Mean scores were computed for each statement, and the results indicated that all statements received a rating of "Strongly Agree," reflecting a high level of agreement. The overall mean score across all statements was 3.56, further reinforcing the "Strongly Agree" rating.

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSION, AND RECOMMENDATION

5.1 Summary of Findings

The project SWAMP aims to develop an early warning system called the Flood Monitoring, Prediction, and Alert System. It focuses on providing timely information and alerts to the residents of Barangay Frances. The system includes a webcam that continuously monitors water levels and streams live footage to a dedicated website. Various sensors, including those for water level, precipitation rate, and atmospheric conditions, are utilized to predict flood levels up to three hours in advance. The website serves as a central platform to display the overall system, including notification and alert features.

To ensure accuracy, the hydrological sensors were deployed and tested at different locations within Barangay Frances. Sensors such as ultrasonic sensor, barometric pressure sensor, and rain gauge sensor are utilized to gather data which will then be used for monitoring and predictions.

The image processing component of the system was trained using images captured by the camera. The performance of the image processing component of the system is tested by live streaming its feed. It is validated with a confusion matrix and was able to yield 97% accuracy.

The prediction aspect utilized data provided by PAGASA for training. The program was able to yield 1.18 percentage error upon testing and validation, indicating that the accuracy of forecasting is significantly high.

5.2 Conclusion

Based on the data gathered and test conducted, the following are the collected conclusions by the proponents:

1. The project operates independently by harnessing power from solar panels and a battery. It also establishes its own internet connection through a portable Wi-Fi, which is connected to the system. The sensors employed in the project effectively gather the required data, which is utilized for the prediction aspect. To cater to the preferences of the barangay, the proponents have developed a modular device for flexible deployment. The system comprises three separate nodes located 150 meters apart, communicating with each other through node MCU and the internet connection.
2. The project incorporates image processing through a webcam to monitor the water level of the Barangay. The YOLOv5 model is utilized for this purpose, and the live stream of the monitoring is made available on the project's website. The predictive model utilizes the data collected from the sensors to anticipate the flood level, providing a three-hour lead time. The results are then classified according to the standards set by PAGASA (Philippine Atmospheric, Geophysical and Astronomical Services Administration).
3. The website serves as a central platform that presents all the essential data gathered from the sensors, including the results of image processing for flood monitoring and the system's predictions. Additionally, the website incorporates an alert system that utilizes real-time data to notify the residents about the current flood level in Barangay. Through the website, residents can access timely information and

updates regarding the flood situation, enabling them to stay informed and take necessary precautions accordingly.

4. The SWAMP device, which was deployed in the barangay, has successfully collected accurate data through its sensors, image processing, and prediction capabilities. The project proponents tailored the device to suit the specific needs of the barangay and conducted a comprehensive evaluation to verify its functionality, reliability, usability, efficiency, maintainability, and portability.

5.3 Recommendation

As the system has been fully deployed, the proponents created the following recommendations to improve the overall project.

1. Perform comprehensive research to identify top-quality sensors specifically designed for flood monitoring applications. Look for sensors with a proven track record of accuracy, reliability, and durability in detecting essential parameters like water levels and rainfall. Consider crucial factors such as the sensor's measurement range, durability, response time, and compatibility with the monitoring system. Thoroughly evaluate available options to ensure the selection of high-grade sensors that meet the project's requirements and can provide accurate and reliable data for effective flood monitoring.
2. To enhance the flood monitoring capabilities, the project incorporates flood monitoring aspects throughout all parts of the node. This includes integrating sensors and systems that can effectively capture and analyze flood-related data in real-time. Additionally, efforts are made to train the

model with a larger dataset to improve its accuracy in detecting and classifying flood-related objects. The project also considers the utilization of newer versions of the YOLO model, which are known for their advancements in object detection capabilities. These enhancements aim to optimize the flood monitoring system and ensure more accurate and reliable results.

3. To enhance the flood monitoring capabilities, the project incorporates flood monitoring aspects throughout all parts of the node. This includes integrating sensors and systems that can effectively capture and analyze flood-related data in real-time. Additionally, efforts are made to train the model with a larger dataset to improve its accuracy in detecting and classifying flood-related objects. The project also considers the utilization of newer versions of the YOLO model, which are known for their advancements in object detection capabilities. These enhancements aim to optimize the flood monitoring system and ensure more accurate and reliable results.
4. The project also includes the exploration of alternative programming languages and models to enhance performance and improve the accuracy of flood level prediction. The project also considers predicting the subsidence of flooding, to further ensure the safety of the residents in returning to their respective properties.

References

Abaya, W. F., Basa, J., Sy, M., Abad, A. C., & Dadios, E. P. (2014). Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV. In 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-6). doi: 10.1109/HNICEM.2014.7016253.

Acharya, R. (2017). Tropospheric impairments: Measurements and mitigation. doi: 10.1016/b978-0-12-809732-8.00007-7.

Akouemo, H. N., & Povinelli, R. J. (2017). Data improving in time series using ARX and Ann Models. IEEE Transactions on Power Systems, 32(5), 3352–3359.

Alizadeh Kharazi, B., & Behzadan, A. H. (2021). Flood depth mapping in street photos with image processing and deep neural networks. Computers, Environment and Urban Systems, 88, 101628.

Bande, S., & Shete, V. V. (2018). Smart flood disaster prediction system using IoT & neural networks. In Proc. 2017 Int. Conf. Smart Technol. Smart Nation, SmartTechCon 2017 (pp. 189–194). doi: 10.1109/SmartTechCon.2017.8358367.

Basu, B., Morrissey, P., & Gill, L. W. (2022). Application of nonlinear time series and machine learning algorithms for forecasting groundwater flooding in a lowland karst area. Water Resources Research, 58(2).

Bello, R. (2021, January 28). Aerotropolis heightens disaster risks for Bulacan communities. Retrieved from Rappler: https://www.rappler.com/environment/aerotropolis-airport-heightensdisaster-risks-communities-bulacan/

Dondang, R. A., Mansor, Z., & Satrya, G. B. (2021). Design and Development of Flood Monitoring and Early Warning System. Malaysian J. Sci. Adv. Technol., 1(3), 72–76.

M, V. G., & Hameed, R. (2020). Iot Based Early Flood Monitoring, Detection and Alarming System. Int. J. Innov. Res. Sci. Eng. Technol., 6(2), 7817. Retrieved from www.ijariie.com1648

Hassan, W. H. W., Jidin, A. Z., Aziz, S. A. C., & Rahim, N. (2019). Flood disaster indicator of water level monitoring system. International Journal of Electrical and Computer Engineering (IJECE), 9(3), 1694–1699. doi: 10.11591/ijece.v9i3.pp1694-1699.

Jain, P., Schoen-Phelan, B., & Ross, R. (2020). Automatic flood detection in sentinel-2 images using deep convolutional neural networks. In Proceedings of the 35th Annual ACM Symposium on Applied Computing.

P. | Arduino | 5. (2017, May 29). How to Set Up the BMP180 Barometric Pressure Sensor on an Arduino. Circuit Basics. Retrieved from https://www.circuitbasics.com/set-bmp180-barometric-pressure-sensor-arduino/

Kim, J., et al. (2021). Synergistic use of geospatial data for water body extraction from sentinel-1 images for operational flood monitoring across southeast Asia using deep neural networks. Remote Sens., 13(23), 1–22. doi: 10.3390/rs13234759

Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I., & Baba, D. (2019). Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. Water, 12(1), 96.

Kovantsev, A., & Gladilin, P. (2020). Analysis of multivariate time series predictability based on their features. In 2020 International Conference on Data Mining Workshops (ICDMW) (pp. 348–354).

Kostadinov, S. (n.d.). Understanding GRU networks - towardsdatascience.com. Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

Kumar, S., Hussain, L., Banarjee, S., & Reza, M. (2018). Energy load forecasting using Deep Learning Approach-LSTM and GRU in spark cluster. In 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT).

Lady Ada. (2021). Force Sensitive Resistor (FSR) Created by lady ada (p. 19). Retrieved from https://learn.adafruit.com/force-sensitive-resistor-fsr

Le, V. M., Pham, B. T., Le, T.-T., Ly, H.-B., & Le, L. M. (2020). Daily rainfall prediction using nonlinear autoregressive neural network. Micro-Electronics and Telecommunication Engineering, pp. 213–221.

Lopez-Fuentes, L., Rossi, C., & Skinnemoen, H. (2017). River segmentation for flood monitoring. In Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017 (pp. 3746–3749). doi: 10.1109/BigData.2017.8258373

Mendoza-Cano, O., et al. (2021). Experiments of an IoT Based Wireless Sensor Network for flood monitoring in Colima Mexico.

Moy De Vitry, M. A., Kramer, S., Wegner, J. D., & Leitao, J. P. (2019). Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. Hydrol. Earth Syst. Sci., 23(11), 4621–4634. doi: 10.5194/hess-23-4621-2019

Noor, M., Ibrahim, B., Hisham, A., Mohd, B., & Latib, A. Z. B. (2020). Flood Monitoring and Prediction System: An Implementation of Dynamic Web (PHP) and Internet of Thing (IoT), 1(1), 1–8.

Patil, S., Pisal, J., Patil, A., Ingavale, S., Ayarekar, P., & Mulla, S. (2019). IRJET-A Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks A Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks. Int. Res. J. Eng. Technol., p. 1807. Retrieved from www.irjet.net

Purkovic, D., et al. (2019). Smart River Monitoring and Early Flood Detection System in Japan Developed with the EnOcean Long Range Sensor Technology.

Rani, D. S., Jayalakshmi, G. N., & Baligar, V. P. (2020). Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction. In 2nd Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2020 - Conf. Proc. (pp. 261–267). doi: 10.1109/ICIMIA48430.2020.9074928

Sahagun, M. A. M., Cruz, J. C. D., & Garcia, R. G. (2017). Wireless sensor nodes for flood forecasting using artificial neural network. In HNICEM 2017 - 9th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. (Vol. 2018-Janua, pp. 1–6). doi: 10.1109/HNICEM.2017.8269462

Sakib, S. N., Ane, T., Matin, N., & Kaiser, M. S. (2016). An intelligent flood monitoring system for Bangladesh using wireless sensor network. Sensors, 21(16), 5614.

Satria, D., Yana, S., Munadi, R., & Syahreza, S. (2018). Design of Information Monitoring System Flood Based Internet of Things (IoT). In Proceedings of MICoMS 2017 (pp. 337–342). doi: 10.1108/978-1-78756-793-1-00072

Saxena, S. (2021, March 18). LSTM: Introduction to LSTM: Long short term memory. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

Sensortec, B. (2014). BMP085 digital pressure sensor (p. 27). Retrieved from https://www.sparkfun.com/datasheets/Components/General/BST-BMP085-DS000-05.pdf

Shah, W., et al. (2018). The Implementation of IoT Based Flood Monitoring and Alert System.

Shih, S.-Y., Sun, F.-K., & Lee, H.-y. (2019). Temporal pattern attention for multivariate time series forecasting. Machine Learning, 108(8-9), 1421–1441.

Simatupang, J. W., & Naufal, F. (2019). Flood Early Warning Detection System Prototype Based on IoT Network.

Song, Z., & Tuo, Y. (2021). Automated flood depth estimates from online traffic sign images: Explorations of a convolutional neural network-based method. Sensors, 21(16), 5614.

Subashini, M. J. (2021). Early Warning System using Weather Forecasting Data and Wireless Sensor Networks-A Review.

Thekkil, T. M., & Prabakaran, N. (2017). Real-time WSN based early flood detection and control monitoring system. In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). doi: 10.1109/icicict1.2017.8342828

Tolentino, L. K. S., et al. (2019). Real Time Flood Detection, Alarm and Monitoring System Using Image Processing and Multiple Linear Regression.

Maker Portal. (2020, May 26). Capacitive Soil Moisture Sensor Calibration with Arduino. Retrieved from https://makersportal.com/blog/2020/5/26/capacitive-soil-moisture-calibration-with-arduino#:~:text=Calibration%20Procedure%3A&text=Measure%20mass%20of%20the%20dry

Chm. (2018, September 4). Introduction to 1d convolutional neural networks in Keras for time sequences. LaptrinhX. Retrieved from https://laptrinhx.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3063469381/

DCDRRMO | Disaster Risk Reduction and Management. (n.d.). Retrieved from https://drrmo.davaocity.gov.ph/

Official Gazette of the Republic of the Philippines. (2020). How to make sense of PAGASA's color-coded rainfall advisories | GOVPH. Retrieved from https://www.officialgazette.gov.ph/how-to-make-sense-of-pagasas-color-coded-warning-signals/

Future Learn. (2021, November 17). How the Philippines deals with typhoons. Retrieved from https://www.futurelearn.com/info/futurelearn-international/philippinestyphoons#:~:text=winds%20and%20rain.,Why%20is%20the%20Philippines%20prone%20to%20typhoons%3F,roughly%2020%20typhoons%20each%20year

Instructables. (2019, January 22). Arduino UNO: Complete Ultrasonic Sensor Guide (HC-SR04). Retrieved from https://www.instructables.com/Arduino-UNO-Complete-Ultrasonic-Sensor-Guide-HC-SR/

Intek, Inc. (n.d.). Low Flow Liquid and Water Flow Meters. Retrieved from https://intekflow.com/flow-measurement-instruments/low-flow-meters-liquid/

OmniVision Technologies Inc. (2009). OV5647 datasheet (pp. 6-1).

PAGASA. (n.d.). About Tropical Cyclones. Retrieved from https://www.pagasa.dost.gov.ph/information/about-tropical-cyclone

GeeksforGeeks. (2021, June 25). Understanding of LSTM networks. Retrieved from https://www.geeksforgeeks.org/understanding-of-lstm-networks/#:~:text=LSTMs%20provide%20us%20with%20a,)%2C%20which%20is%20an%20advantage.

Electronic Clinic. (2019, October 6). Water Flow Sensor Arduino, Water Flow Rate & Volume Measurement. Retrieved from https://www.electroniclinic.com/water-flow-sensor-arduino-water-flow-rate-volume-measurement/

ES Systems. (2021, February 22). What is Barometric Pressure and how is it measured? Retrieved from https://esenssys.com/what-is-barometric-pressure/

ANNEX I

BILL OF MATERIALS

BILL OF MATERIALS

Item #	Unit	Item Description	Qty.	Unit Cost	Total Cost
1	pc	Raspberry Pi 4B	1	9,000.00	9,000.00
2	pc	Raspberry Pi Cam	1	1,070.00	1,070.00
3	pc	Solar Panel 80W	1	-	-
4	pc	Solar Panel 10W	2	-	-
5	pc	Solar Panel 35W	3	1,400.00	2,800.00
6	pc	Solar Charge Controller	4	220.00	660.00
7	pc	Inverter (12V to 220V)	1	1,300.00	1,300.00
8	pc	Battery (25AH, 12V)	1	1,599.00	1,599.00
9	pc	Battery (7AH, 12V)	2	490.00	980.00
10	pc	CSB Battery (9AH, 12V)	2	800.00	1,600.00
11	pc	Panel Box (500x400x180 mm)	1	1,750.00	1,750.00
12	pc	Panel Box (400x300x180 mm)	2	1,450.00	2,900.00
13	pc	NodeMCU (Makerlab)	10	160.00	1,600.00
14	pc	NodeMCU (Deeco)	3	200.00	600.00
15	pc	Water Flow Meter	1	250.00	250.00
16	pc	BMP180	2	40.00	80.00
17	pc	Rain Gauge (Makerlab)	1	1,300.00	1,300.00
18	pc	Rain Gauge (Shopee)	2	845.00	1,690.00
19	pc	Ultrasonic Sensor (HC-SR04)	3	65.00	195.00
20	pc	Ultrasonic Sensor (waterproof)	3	290.00	870.00
21	pc	Pocket Wifi	1	200.00	200.00
22	pc	Pocket Wifi	3	350.00	1,050.00

23	pc	Hall Effect Sensor	3	50.00	150.00
24	pc	Webcam	1	850.00	850.00
25	pc	GI Steel Pipe #2	2	1,880.00	3,760.00
26	pc	GI Steel Pipe #2	1	1,750.00	1,750.00
27	pc	Coupling (straight)	2	130.00	260.00
28	pc	Coupling (elbow)	2	180.00	360.00
29	pc	Coupling (tee)	5	240.00	1,200.00
30	pc	Coupling #2	2	170.00	340.00
31	m	Angle bar	1	320.00	320.00
32	m	Flat bar	1	220.00	220.00
33	pc	Tires	1	100.00	100.00
34	sack	Cement	1	285.00	285.00
35	sack	Cement	1	220.00	220.00
36	kg	Cement	1	100.00	100.00
37	pc	PVC pipe (electrical)	1	175.00	175.00
38	pc	PVC pipe (electrical)	1	180.00	180.00
39	pc	PVC pipe (electrical)	1	130.00	130.00
40	pc	PVC elbow (electrical)	5	30.00	150.00
41	pc	Elbow (1/2)	10	15.00	150.00
42	pc	Tee (1/2)	5	15.00	75.00
43	pc	PVC elbow (electrical)	8	30.00	240.00
44	pc	Blue pipe (1/2)	1	85.00	85.00
45	pc	Elbow blue (1/2)	10	20.00	200.00
46	pc	Elbow (1/2)	10	10.00	100.00

47	pc	Elbow orange (3/4)	5	20.00	100.00
48	pc	Tee (3/4)	5	30.00	150.00
49	pc	Bisagra (small)	4	15.00	60.00
50	pc	Neltex 100cc	1	100.00	100.00
51	pc	Hose clear (1/2)	1	25.00	25.00
52	pc	Elbow (3/4)	5	20.00	100.00
53	pc	Tee (3/4)	5	30.00	150.00
54	pc	Elbow #1	5	40.00	200.00
55	pc	Tee #1	5	45.00	225.00
56	pc	Coupling #1	6	40.00	240.00
57	pc	Acrylic Sheet	1	600.00	600.00
58	pc	Threading	16	150.00	2,400.00
59	-	Labor	1	2,400.00	2,400.00
60	-	Electricity	1	500.00	500.00
61	pc	Black screw	20	2.00	40.00
62	pc	Washer	20	2.00	40.00
63	pc	Bolt & screw #10	4	12.00	48.00
64	pc	Job screw bolt	16	7.00	112.00
65	pc	Bolt & knots	8	9.00	72.00
66	pc	Stove bolt	10	8.00	80.00
67	pc	Washer	10	1.00	10.00
68	pc	Bolt knot	10	10.00	100.00
69	pc	Washer	10	10.00	100.00
70	pc	Drill bit	1	95.00	95.00

71	pc	Stove bolt	6	7.00	42.00
72	pc	Bolt & screw #12	6	12.00	72.00
73	pc	Tox & screw	8	3.00	24.00
74	pc	Bolt screw	4	15.00	60.00
75	m	Micro USB to USB (wire)	10	14.00	140.00
76	m	Wires (M to F)	2	70.00	140.00
77	m	Wires (22AWG)	12	5.00	60.00
78	m	Cord (speaker wires)	25	17.00	425.00
79	m	Royal cable (16/3)	8	70.00	560.00
80	m	Royal cable (16/4)	16	105.00	1,680.00
81	m	Micro USB cable	2	118.14	236.28
82	pc	Servo motor	1	90.00	90.00
83	pc	USB port	3	70.00	210.00
84	pc	Breadboard	8	50.00	400.00
85	pc	Cable ties	10	20.00	200.00
86	pc	Paint brush	1	35.00	35.00
87	pc	Masking tape	1	35.00	35.00
88	pc	Epoxy	1	130.00	130.00
89	pc	Cutting disk	1	35.00	35.00
90	pc	Teflon	3	15.00	45.00
91	pc	Paleta	1	30.00	35.00
92	pc	Solvent (neltex)	1	103.00	103.00
93	pc	Butane torch	1	235.00	235.00
94	pc	Teflon	5	15.00	75.00

95	pc	Cutting disk	1	50.00	50.00
96	pc	Grinding disk	1	35.00	35.00
97	pc	Cutting disk	2	25.00	50.00
98	pc	Teflon	3	30.00	90.00
99	pc	Drill bit	1	180.00	180.00
100	pc	Wifi module	6	110.00	660.00
101	L	Paint thinner	1	50.00	50.00
102	pc	Paint brush	2	25.00	50.00
103	pc	Liston	3	40.00	120.00
104	pc	Paint brush	1	30.00	30.00
105	pc	Paleta	1	15.00	15.00
106	pack	Cable ties	1	88.00	88.00
107	pc	Plastic container	1	88.00	88.00
108	pc	Epoxy	1	145.00	145.00
109	pc	Mighty bond	1	60.00	60.00
110	pc	Drill bit (1/4)	1	140.00	140.00
111	pc	Liha #100	2	20.00	40.00
112	pack	Magnets	1	40.00	40.00
113	pc	Metal spreaders	1	40.00	40.00
114	L	Paint	4	80.00	320.00
115	pc	Row plug holder	20	10.00	200.00
116	pc	Mounting tape	1	400.00	400.00
117	pc	3M tape	1	700.00	700.00
118	pack	Cable ties	1	20.00	20.00
				TOTAL	**59,514.28**

ANNEX II

EVALUATION FORMS

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: Rose ann Quiantao Petsa: ______
Address: Francer, Calumpit, Bulacan Kasarian: Female Edad: 30
Contact Number: 0965-647-0398 E-mail: roseannquiantao26@gmail.com

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*			✓	
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*	✓			
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*			✓	
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*			✓	
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*			✓	
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*			✓	
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	[signature]

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*			✓	
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*			✓	
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*			✓	

Katanungan o Komento:

SIGNATURE (LAGDA):

Transaction ID	
Signature	

		Index No.	F-UES-4.2-CPP
	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: Nerizel G. Torres
Petsa: 6-13-23
Address: Purok 5, Francis, Cal, Bul Kasarian: BABAE Edad: 21
Contact Number: 09117532687 E-mail: batangtondo@gmail.com

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*	~~✓~~			✓
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*		X		
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*	~~✓~~			✓
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*	~~✓~~			✓
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*			✓	
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*			✓	
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)			✓	
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*			✓	
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*		✓		
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*			✓	

Katanungan o Komento:

In this case maganda rin na may mga ganito tayong device. Nang sa gayun ay ma aware tayo sa kung anong pwedeng mangyare sa lugar naten, at mas maagap tayong maglikas.

SIGNATURE (LAGDA):

Reizel Jerrera

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: Cherryl B. Pino Petsa: 6-13-'23
Address: ______ Kasarian: F Edad: 42
Contact Number: ______ E-mail: ______

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*				✓
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*		✓		~~✓~~
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*				✓
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*				✓
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*				✓
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*				✓
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*				✓
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*				✓
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*				✓
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*				✓
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*				✓
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*				✓
2.	Is the authentication of users is easily processed?				

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				✓
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				✓
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*				✓
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*				✓
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*				✓
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*				✓
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*				✓
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				✓

Katanungan o Komento:

____

____

____

____

____

SIGNATURE (LAGDA):

Cherryl B. Pru

Transaction ID	
Signature	

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
	Issue No.	01
	Revision No.	00
	Date	11242017
VRE-UES — **SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
	QAC No.	CC-11242017

Pangalan: Jenny Mundo Petsa: June 17, 2023
Address: Francis Calumpit Bulacan Kasarian: F Edad: 38
Contact Number: 09105831907 E-mail: jennymundo110@gmail.com

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

		Higit na hindi sumasang-ayon	Hindi sumasa ng-ayon	Sumasa ng-ayon	Higit na sumasa ng-ayon
	Katangian	**1**	**2**	**3**	**4**
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*			✓	
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*		✓		
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*			✓	
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*			✓	
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*			✓	
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*			✓	
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*			✓	
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*			✓	
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*			✓	

Katanungan o Komento:

____

____

____

____

SIGNATURE (LAGDA):

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)			✓	
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				✓
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*				✓
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*		✓		
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				✓

Katanungan o Komento:

____

____

____

____

____

SIGNATURE (LAGDA):

Transaction ID	
Signature	

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES
Ayala Blvd., Ermita, Manila, 1000, Philippines
Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063
Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph

Index No.	F-UES-4.2-CPP
Issue No.	01
Revision No.	00
Date	11242017
Page	1 / 1
QAC No.	CC-11242017

VRE-UES

SWAMP PROJECT EVALUATION FORM

Pangalan: ARLENE M. LAPERA Petsa: JUNE 13.2023
Address: PUROK 5 FRANCEZ CALUMPIT BULACAN Kasarian: BABAE Edad: 43
Contact Number: 09359614214 E-mail: arlenemunda08@gmail.com

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*			✓	
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*				✓
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*			✓	
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*			✓	
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*			✓	
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*			✓	
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*				✓
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*			✓	
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*		✓	✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*		✓		
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*			✓	

Katanungan o Komento:

ANONG PAGBABAGO ANG MAAARING IDULOT SA LUGAR NAMIN NG INYONG PROYEKTO?

SIGNATURE (LAGDA): arnSapera

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: Candy Garcia Petsa: ______
Address: Francisco Calumpit Bulacan Kasarian: FM Edad: 31
Contact Number: 09957013030 E-mail: ______

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*				✓
2.	Will it not cause inconvenience upon use? *(Ito ba hindi ay nag-dudulot ng abala?)*				✓
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*				✓
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*				✓
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*				✓
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*				✓
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*				✓
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*				✓
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?				✓

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				✓
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*				✓
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*				✓
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*			✓	
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*				✓
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				✓

Katanungan o Komento:

SIGNATURE (LAGDA):

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: MANNY MAHILING Petsa: 06/14/23
Address: P-J BRGY FRANCES CALUMPIT BULACAN Kasarian: M Edad: 34
Contact Number: ______________________ E-mail: ______________________

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*			✓	
2.	Will it not cause inconvenience upon use? *(Ito ba hindi ay nag-dudulot ng abala?)*			✓	
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*			✓	
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*			✓	
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*			✓	
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*			✓	
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*			✓	
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*			✓	
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph		Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)			✓	
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*			✓	
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*			✓	
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*			✓	

Katanungan o Komento:

SIGNATURE (LAGDA):

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	**SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
		QAC No.	CC-11242017

Pangalan: Dan Kenneth M. Makiling Petsa: 06/14/02
Address: PUROK 5 FRANCES CALUMPIT BULACAN Kasarian: Male Edad: 20
Contact Number: 09207171714 E-mail: kmakiling3@gmail.com

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*			✓	
2.	Will it not cause inconvenience upon use? *(Ito ba ay hindi nag-dudulot ng abala?)*				✓
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*			✓	
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*				✓
	RELIABILITY				
1.	Does the machine operate property most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*			✓	
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*				✓
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*				✓
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*			✓	
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*				✓
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*				✓
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*				✓
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*			✓	
2.	Is the authentication of users is easily processed?				

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				✓
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				✓
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*				✓
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*		✓		
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*			✓	
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				✓

Katanungan o Komento:

____

____

____

____

____

SIGNATURE (LAGDA):

Transaction ID	
Signature	

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
	Issue No.	01
	Revision No.	00
	Date	11242017
VRE-UES — **SWAMP PROJECT EVALUATION FORM**	Page	1 / 1
	QAC No.	CC-11242017

Pangalan: Aurora Sentero Petsa: 6/14/23
Address: Frances Bulacan Kasarian: F Edad: 69
Contact Number: - E-mail: -

Ang mga sumusunod na katanungan ay nagsisilbing pagsusuri para sa proyektong ***SWAMP: FLOOD MONITORING, PREDICTION, AND ALERT SYSTEM USING WIRELESS SENSOR NETWORK AND INTERNET OF THINGS (IoT)***. Maaaring suriin ang mga sumusunod na katanungan at timbangin mula 1-4 base sa iyong opinion. Maglagay lamang ng **check (✓)** sa kahon. Maaaring maglagay din ng katanungan at komento patungkol sa nasabing proyekto.

	Katangian	Higit na hindi sumasang-ayon 1	Hindi sumasa ng-ayon 2	Sumasa ng-ayon 3	Higit na sumasa ng-ayon 4
	FUNCTIONALITY				
1.	Does the machine perform the task required? *(Nagagawa ba ng device nang maayos ang layunin nito?)*				✓
2.	Will it not cause inconvenience upon use? *(Ito ba ay nag-dudulot ng abala?)*				✓
3.	Does the machine respond correctly based on its purpose upon user interaction? *(Nagagamit ba nang maayos ang device base sa nasabing layunin nito?)*				✓
4.	Will it operate over time? *(Ito ba nagagamit nang mahabang panahon?)*				✓
	RELIABILITY				
1.	Does the machine operate properly most of the time? *(Madalas bang gumagana ba ng maayos at walang aberya ang device?)*				✓
2.	Does the authentication of users correctly process each user's info? *(Tama bang pinoproseso ng pagpapatunay ng mga user ang impormasyon ng bawat user?)*			✓	
3.	Does the machine open for revisions once a negative impact will occur? *(Bukas ba ang device para sa mga rebisyon sa sandaling magkaroon ng negatibong epekto?)*				✓
4.	Does the machine be able to detect any possible malfunction? *(Nakikita ba ng device ang anumang posibleng problema nito?)*				✓
	USABILITY				
1.	Was it user-friendly and easy to operate? *(Ito ba ay madaling intindihin at madaling gamitin?)*				✓
2.	Does it have manuals for easy instructions? *(Mayroon ba itong mga manwal para sa madaling mga panuto?)*				✓
3.	Does the system suit the targeted users in the given environment? *(Nababagay ba ang system sa mga naka-target na user sa ibinigay na kapaligiran?)*				✓
	EFFICIENCY				
1.	Does the system monitors in real-time? *(Nasusubaybayan ba nang real-time ang lagay ng nangyayari sa kapaligiran na may kinalaman sa proyekto?)*				✓
2.	Is the authentication of users is easily processed?			✓	

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 106 | Fax No. +632-521-4063 Email: vpaa@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-UES-4.2-CPP
		Issue No.	01
		Revision No.	00
		Date	11242017
VRE-UES	SWAMP PROJECT EVALUATION FORM	Page	2 / 1
		QAC No.	CC-11242017

	(Madali bang i-proseso ang impormasyon ng mga gumagamit nito?)				
	MAINTAINABILITY				
1.	Does the system easily find a failure when it occurs? *(Madali bang natutukoy ng system kung mayroon man itong problema?)*				✓
2.	Does it modify instantly when a problem arises? *(Ito ba ay madaling ayusin kung mayroon mang problema?)*			✓	
3.	Can the system avoid unexpected effects from the modifications? *(Kaya ba ng system na iwasan ang hindi inaasahang epekto ng mga pagbabago nito?)*			✓	
4.	Does the sensors work properly? *(Gumagana ba nang maayos ang mga sensors na ginamit dito?)*			✓	
	PORTABILITY				
1.	Can the machine be moved to other environments? *(Kaya bang ilipat-lipat ng lugar ang device?)*		✓		
2.	Can the machine be able to adopt changes in technology when in need of an update? *(Kaya ba ng device na umangkop sa anumang pagbabago ng teknolohiya?)*				✓
3.	Does the machine able to comply with program quality standards? *(Natutugunan ba ng device ang mga naaayon na pamantayan sa uri ng proyekto?)*				✓

Katanungan o Komento:

A. M. Cortez

SIGNATURE (LAGDA):

Transaction ID	
Signature	

ANNEX III

SWAMP CODES

Sensors Codes

Ultrasonic Sensor

```
#include <ESP8266WiFi.h>

String apiKey = "V72YHRUPPGGIBPI0"; // write your "Write API key"
const char* ssid = "SeeFlood(SWAMP)"; // write your "wifi name"
const char* password = "Swamp2022"; // write your "wifi password"
const char* server = "api.thingspeak.com";
WiFiClient client;

//for NodeMCU
const int trigPin = 12;
const int echoPin = 13;
// defines variables
long duration;
int distance;

void setup()
{
  Serial.begin(9600);
  Serial.println("Serial Begin");
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);
  WiFi.begin(ssid, password);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

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

void loop()
{
  ultra();
  delay(1000);

  // we have made changes in program to decrease the latency.
  // but it will take minimum 15 second to post data on Thingspeak channel.
    if (client.connect(server,80))
    {
      fwd_to_Thingspeak();
    }
    client.stop();
    Serial.println("Waiting");
    delay(1000);
}

void ultra()
{
```

```
// number of measurements to average
const int numReadings = 20;

static int readings[numReadings];      // the readings from the ultrasonic sensor
static int index = 0;                  // the index of the current reading
static int total = 0;                  // the total of the readings

// remove the oldest reading from the total
total -= readings[index];

// take a new reading from the ultrasonic sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = 290 - (290 - (duration * 0.034 / 2));

// add the new reading to the total
readings[index] = distance;
total += readings[index];

// advance to the next index in the buffer
index++;
if (index >= numReadings) {
  index = 0;
}
```

```
  // calculate the average of the readings
  distance = total / numReadings;

  // print the distance to the serial monitor
  Serial.println("Distance = " + String(distance) + "cm");
}

void fwd_to_Thingspeak()
{
  String postStr = apiKey;
  postStr +="&field1=";
  postStr += String(distance);  // ultrasonic data
  postStr += "\r\n\r\n";

  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);

  Serial.println("Send data to thingspeak: ");
  Serial.print("Content-Length: ");
  Serial.print(postStr.length());
```

```
  Serial.print(" Field-1: ");
  Serial.print(distance);  // ultrasonic data
  Serial.println(" data send");
}
```

Barometric Pressure Sensor

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>

// WiFi network credentials
const char* ssid = "SeeFlood(SWAMP)";
const char* password = "Swamp2022";

// ThingSpeak API key
const char* apiKey = "7AI1BCLDQKT69NNH";

// Initialize BMP180 sensor
Adafruit_BMP085 bmp;

// Initialize WiFi client
WiFiClient client;

void setup() {
```

```
  Serial.begin(9600);

  // Connect to WiFi network
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Initialize ThingSpeak library
  ThingSpeak.begin(client);

  // Initialize BMP180 sensor
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP180 sensor, check wiring!");
    while (1) {}
  }
}

void loop() {
  // Read atmospheric pressure from BMP180 sensor
  float pressure = bmp.readPressure() / 100.0F;
  Serial.print("Pressure = ");
  Serial.print(pressure);
  Serial.println(" hPa");

  // Send data to ThingSpeak
```

```
  ThingSpeak.setField(1, pressure);
  int response = ThingSpeak.writeFields(1, apiKey);
  if (response == 200) {
    Serial.println("Data sent to ThingSpeak successfully");
  } else {
    Serial.println("Error sending data to ThingSpeak");
  }

  // Wait 15 seconds before sending next data
  delay(15000);
}
```

Rain Gauge

```
#include <ESP8266WiFi.h>
#define RAIN_PIN 0    // interrupt pin
#define CALC_INTERVAL 1000  // increment of measurements
#define DEBOUNCE_TIME 15    // time * 1000 in microseconds required to get through bounce noise
#define ML_PER_TIP 5      // number of ml per tip

// http://texaselectronics.com/media/mconnect_uploadfiles/t/r/tr-525i_rainfall_user_s_manual.pdf
// Per manufatures spec on bucket being tested:

// "Average Switch closure time is 135 ms"
// "Bounce Settling Time: 0.75 ms"

String apiKey = "DU32KYMBNWACTQ6B"; // write your "Write API key"
```

```
const char* ssid = "SeeFlood(SWAMP)"; // write your "wifi name"
const char* password = "Swamp2022"; // write your "wifi password"
const char* server = "api.thingspeak.com";
WiFiClient client;



unsigned long nextCalc;
unsigned long timer;
unsigned long lastResetTime; // added variable to track last time count was reset

volatile unsigned int rainTrigger = 0;
volatile unsigned long last_micros_rg;

void setup() {
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(RAIN_PIN), countingRain, RISING);

  pinMode(RAIN_PIN, INPUT);
  nextCalc = millis() + CALC_INTERVAL;
  lastResetTime = millis(); // set initial value for lastResetTime

  WiFi.begin(ssid, password);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  while (WiFi.status() != WL_CONNECTED)
  {
```

```
    delay(500);
    Serial.print(".");
  }
    Serial.println("");
    Serial.println("WiFi connected");
}


void loop()
{
  ultra();
  delay(1000);
  // we have made changes in program to decrease the latency.
  // but it will take minimum 15 second to post data on Thingspeak channel.
    if (client.connect(server,80))
    {
      fwd_to_Thingspeak();
    }
    client.stop();
    Serial.println("Waiting");
    delay(1000);
}


void ultra() {
  timer = millis();
  if(timer > nextCalc) {
    nextCalc = timer + CALC_INTERVAL;
```

```
    Serial.print("Total Tips: ");
    Serial.print(rainTrigger);
    Serial.print("  Total Volume: ");
    Serial.print(rainTrigger * ML_PER_TIP);
    Serial.println(" ml");
  }

  // reset count every minute
  if(timer >= lastResetTime + 60000) {
    rainTrigger = 0;
    lastResetTime = timer;
  }
}

ICACHE_RAM_ATTR void countingRain() {
  // ATTEMPTED: Check to see if time since last interrupt call is greater than
  // debounce time. If so, then the last interrupt call is through the
  // noisy period of the reed switch bouncing, so we can increment by one.
  if((long)(micros() - last_micros_rg) >= DEBOUNCE_TIME * 1000) {
   rainTrigger += 1;
   last_micros_rg = micros();
  }
}


void fwd_to_Thingspeak()
{
  // Convert total tips to milliliters
```

```
  String postStr = apiKey;
  postStr +="&field1=";
  postStr += String(rainTrigger * ML_PER_TIP);  // send total tips in milliliters
  postStr += "\r\n\r\n";

  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);

  Serial.println("Send data to thingspeak: ");
  Serial.print("Content-Length: ");
  Serial.print(postStr.length());
  Serial.print(" Total tips: ");
  Serial.print(rainTrigger * ML_PER_TIP);
  Serial.print(" Total ml: ");
  Serial.print(rainTrigger * ML_PER_TIP);
  Serial.println(" in");
  Serial.println(" data send");
}
```

Predictive Model Codes

Core

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.preprocessing.sequence import TimeseriesGenerator
from sklearn. preprocessing import MinMaxScaler, StandardScaler
import tensorflow as tf
import csv

mpl.rcParams ['figure.figsize'] = (10, 8)
mpl.rcParams ['axes.grid'] = False

data = pd.read_csv('swamp.csv')

data['Date Time'] = pd.to_datetime(data['Date Time'], infer_datetime_format=True)
data.set_index('Date Time')[['FL', 'RF', 'AP']].plot(subplots=True)

data_input = data[['FL', 'RF', 'AP']]

scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data_input)

features = data_scaled
target = data_scaled[:,0]

TimeseriesGenerator(features, target, length=2, sampling_rate=1, batch_size=1)[0]
```

```
x_train, x_test, y_train, y_test = train_test_split(features, target, test_size=0.20,
random_state=123, shuffle=False)


window_length = 720

batch_size = 32

number_features = 3

train_gen = TimeseriesGenerator(x_train, y_train, length=window_length,
sampling_rate=1, batch_size=batch_size)

test_gen = TimeseriesGenerator(x_test, y_test, length=window_length, sampling_rate=1,
batch_size=batch_size)


model = tf.keras.Sequential()

model.add(tf.keras.layers.LSTM(128, input_shape = (window_length, number_features),
return_sequences=True))

model.add(tf.keras.layers.LeakyReLU(alpha=0.5))

model.add(tf.keras.layers.LSTM(128, return_sequences=True))

model.add(tf.keras.layers.LeakyReLU(alpha=0.5))

model.add(tf.keras.layers.Dropout(0.3))

model.add(tf.keras.layers.LSTM(64, return_sequences=False))

model.add(tf.keras.layers.Dropout(0.3))

model.add(tf.keras.layers.Dense(1))


early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',

                                        patience=2,

                                        mode='min')


model.compile(loss=tf.losses.MeanSquaredError(),

          optimizer=tf.optimizers.Adagrad(learning_rate=0.01),

          metrics=[tf.metrics.MeanAbsoluteError()])
```

```
history = model.fit(train_gen, epochs=20,
                    validation_data=test_gen,
                    shuffle=False,
                    callbacks=[early_stopping])

model.evaluate(test_gen, verbose=0)
model.save('model.h5')

predictions = model.predict(test_gen)

predicted_data = pd.concat([pd.DataFrame(predictions),
pd.DataFrame(x_test[:,1:][window_length:])], axis=1)

reverse_transform = scaler.inverse_transform(predicted_data)

final_prediction =  data_input[predictions.shape[0]*-1:]
final_prediction['FL_PREDICTED']=reverse_transform[:,0]

predicted_values = final_prediction["FL_PREDICTED"]

print(predicted_values)

predicted_values.to_csv("predicted_values.csv", index=False)
```

Classifier

```
import time
import matplotlib.pyplot as plt
```

```
while True:
    import pandas as pd
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
    from PIL import Image

    # Load the dataset
    dataset = pd.read_csv('MAYtraining.csv')  # Replace 'testing.csv' with your actual dataset file

    # Split the data into features (X) and labels (y)
    X = dataset[['FL_PREDICTED']]
    y = dataset['Danger Level']

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create the Random Forest classifier
    classifier = RandomForestClassifier(n_estimators=100, random_state=42)

    # Train the classifier
    classifier.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = classifier.predict(X_test)

    # Evaluate the classifier
    accuracy = accuracy_score(y_test, y_pred)
```

```
print(f'Accuracy: {accuracy}')

# Read the predicted data point from the CSV file
latest_data = pd.read_csv('predicted_values.csv').tail(1)
latest_data = latest_data[['FL_PREDICTED']]  # Remove 'Danger Level' column

# Predict the danger level for the latest data point
prediction = classifier.predict(latest_data)
print(f'Predicted danger level: {prediction[0]}')

def display_color_output(prediction):
    if prediction == "Yellow (Monitor)":
        image_path = "yellow.jpg"  # Path to the yellow image file
    elif prediction == "Orange (Alerto)":
        image_path = "orange.jpg"  # Path to the orange image file
    elif prediction == "Red (Lumikas)":
        image_path = "red.jpg"  # Path to the red image file
    else:
        print("Invalid color")

    # Load and display the image
    image = Image.open(image_path)
    image.show()

display_color_output(prediction)

# Delay for 1 minute
time.sleep(10)
```

Runner

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.preprocessing.sequence import TimeseriesGenerator
from sklearn.preprocessing import MinMaxScaler, StandardScaler
import tensorflow as tf
import time
import os
from googleapiclient.discovery import build
from google.oauth2 import service_account
from googleapiclient.http import MediaFileUpload

credentials =
service_account.Credentials.from_service_account_file('PredictiveAlgo.json')
drive_service = build('drive', 'v3', credentials=credentials)

# Track the ID of the uploaded file
uploaded_file_id = None

scaler = MinMaxScaler()
window_length = 20
batch_size = 5
number_features = 3

# Load the original model architecture
model = tf.keras.Sequential()
```

```
model.add(tf.keras.layers.LSTM(128, input_shape=(window_length, number_features),
return_sequences=True))
model.add(tf.keras.layers.LeakyReLU(alpha=0.5))
model.add(tf.keras.layers.LSTM(128, return_sequences=True))
model.add(tf.keras.layers.LeakyReLU(alpha=0.5))
model.add(tf.keras.layers.Dropout(0.3))
model.add(tf.keras.layers.LSTM(64, return_sequences=False))
model.add(tf.keras.layers.Dropout(0.3))
model.add(tf.keras.layers.Dense(1))

# Load the trained model weights
model.load_weights('model.h5')

while True:
    data = pd.read_csv('FL.csv')  # Replace 'new_data.csv' with the filename/path of your
new dataset

    data['Date Time'] = pd.to_datetime(data['Date Time'], infer_datetime_format=True)
    data.set_index('Date Time')[['FL', 'RF', 'AP']].plot(subplots=True)

    data_input = data[['FL', 'RF', 'AP']]

    # Apply the same scaling as in the original program
    data_scaled = scaler.fit_transform(data_input)

    features = data_scaled

    # Generate time series samples
```

```
    data_gen = TimeseriesGenerator(features, features[:, 0], length=window_length,
sampling_rate=1, batch_size=1)

    # Use the loaded model to predict flood levels
    predictions = model.predict(data_gen)

    predicted_data = pd.concat([pd.DataFrame(predictions),
pd.DataFrame(features[window_length:, 1:])], axis=1)

    # Reverse-transform the predicted values
    reverse_transform = scaler.inverse_transform(predicted_data)

    final_prediction = data_input[predictions.shape[0] * -1:]
    final_prediction['FL_PREDICTED'] = reverse_transform[:, 0]

    predicted_values = final_prediction["FL_PREDICTED"]

    plt.figure()
    plt.plot(predicted_values.values, label='Predicted Flood Level')
    plt.xlabel('Date Time')
    plt.ylabel('FL_PREDICTED')
    plt.legend()
    plt.savefig('FL_Graph.jpg')  # Save the graph as a JPEG file in Google Drive
    plt.close()

    # Delete the previously uploaded file, if any
    if uploaded_file_id:
        try:
            # Update the existing file with the new file content
```

```
            media = MediaFileUpload('FL_Graph.jpg', mimetype='image/jpeg',
resumable=True)
            request = drive_service.files().update(fileId=uploaded_file_id,
media_body=media)
            response = None
            while response is None:
                status, response = request.next_chunk()
                if status:
                    print(f"Uploaded {int(status.progress() * 100)}%")
        except:
            print("Error occurred during file update.")


    # Upload the JPG file to Google Drive if there was no existing file
    if not uploaded_file_id:
        file_metadata = {
            'name': 'FL_Graph.jpg',  # Set the desired filename on Google Drive
            'parents': ['1pJJBZdKBljrsFSr8vLYuB-dNA0Uifmcr'],  # Set the ID of the parent
folder on Google Drive
        }
        media = MediaFileUpload('FL_Graph.jpg', mimetype='image/jpeg',
resumable=True)
        request = drive_service.files().create(body=file_metadata, media_body=media)
        response = None
        while response is None:
            status, response = request.next_chunk()
            if status:
                print(f"Uploaded {int(status.progress() * 100)}%")


        # Store the ID of the newly uploaded file
        uploaded_file_id = response['id']
```

```
    print(predicted_values)

    predicted_values.to_csv("predicted_values.csv", index=False)

    time.sleep(10)
```

Flood Monitoring Codes

Codes for Water Level Gauge Detection

```
import argparse
import os
import platform
import sys
from pathlib import Path
import torch

FILE = Path(__file__).resolve()
ROOT = FILE.parents[0]  # YOLOv5 root directory
if str(ROOT) not in sys.path:
    sys.path.append(str(ROOT))  # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd()))  # relative

from models.common import DetectMultiBackend
from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages, LoadScreenshots, LoadStreams
from utils.general import (LOGGER, Profile, check_file, check_img_size, check_imshow, check_requirements, colorstr, cv2,
                           increment_path, non_max_suppression, print_args, scale_boxes, strip_optimizer, xyxy2xywh)
```

```
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, smart_inference_mode


@smart_inference_mode()
def run(
        weights=ROOT / 'best.pt',  # model path or triton URL
        source=ROOT / 'C:\\Users\\Shane\\Desktop\\draft\\yolov5-master\\custom_dataset\\images',  # file/dir/URL/glob/screen/0(webcam)
        data=ROOT / 'C:\\Users\\Shane\\Desktop\\draft\\yolov5-master\\custom_dataset\\dataset.yaml',  # dataset.yaml path
        imgsz=(640, 640),  # inference size (height, width)
        conf_thres=0.25,  # confidence threshold
        iou_thres=0.45,  # NMS IOU threshold
        max_det=1000,  # maximum detections per image
        device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
        view_img=False,  # show results
        save_txt=False,  # save results to *.txt
        save_conf=False,  # save confidences in --save-txt labels
        save_crop=False,  # save cropped prediction boxes
        nosave=False,  # do not save images/videos
        classes=None,  # filter by class: --class 0, or --class 0 2 3
        agnostic_nms=False,  # class-agnostic NMS
        augment=False,  # augmented inference
        visualize=False,  # visualize features
        update=False,  # update all models
        project=ROOT / 'runs/detect',  # save results to project/name
        name='exp',  # save results to project/name
        exist_ok=False,  # existing project/name ok, do not increment
        line_thickness=3,  # bounding box thickness (pixels)
```

```
        hide_labels=False,  # hide labels
        hide_conf=False,  # hide confidences
        half=False,  # use FP16 half-precision inference
        dnn=False,  # use OpenCV DNN for ONNX inference
        vid_stride=1,  # video frame-rate stride
):
    source = str(source)
    save_img = not nosave and not source.endswith('.txt')  # save inference images
    is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
    is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
    webcam = source.isnumeric() or source.endswith('.streams') or (is_url and not is_file)
    screenshot = source.lower().startswith('screen')
    if is_url and is_file:
        source = check_file(source)  # download

    # Directories
    save_dir = increment_path(Path(project) / name, exist_ok=exist_ok)  # increment run
    (save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True)  #
make dir

    # Load model
    device = select_device(device)
    model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
    stride, names, pt = model.stride, model.names, model.pt
    imgsz = check_img_size(imgsz, s=stride)  # check image size

    # Dataloader
    bs = 1  # batch_size
    if webcam:
```

```
        view_img = check_imshow(warn=True)
        dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
        bs = len(dataset)
    elif screenshot:
        dataset = LoadScreenshots(source, img_size=imgsz, stride=stride, auto=pt)
    else:
        dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt, vid_stride=vid_stride)
    vid_path, vid_writer = [None] * bs, [None] * bs

    # Run inference
    model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz))  # warmup
    seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
    for path, im, im0s, vid_cap, s in dataset:
        with dt[0]:
            im = torch.from_numpy(im).to(model.device)
            im = im.half() if model.fp16 else im.float()  # uint8 to fp16/32
            im /= 255  # 0 - 255 to 0.0 - 1.0
            if len(im.shape) == 3:
                im = im[None]  # expand for batch dim

        # Inference
        with dt[1]:
            visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else False
            pred = model(im, augment=augment, visualize=visualize)

        # NMS
```

```
        with dt[2]:
            pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms,
max_det=max_det)

        # Second-stage classifier (optional)
        # pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)

        # Process predictions
        for i, det in enumerate(pred):  # per image
            seen += 1
            if webcam:  # batch_size >= 1
                p, im0, frame = path[i], im0s[i].copy(), dataset.count
                s += f'{i}: '
            else:
                p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)

            p = Path(p)  # to Path
            save_path = str(save_dir / p.name)  # im.jpg
            txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' else
f'_{frame}')  # im.txt
            s += '%gx%g ' % im.shape[2:]  # print string
            gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
            imc = im0.copy() if save_crop else im0  # for save_crop
            annotator = Annotator(im0, line_width=line_thickness, example=str(names))
            if len(det):
                # Rescale boxes from img_size to im0 size
                det[:, :4] = scale_boxes(im.shape[2:], det[:, :4], im0.shape).round()

                # Print results
```

```
    for c in det[:, 5].unique():
        n = (det[:, 5] == c).sum()  # detections per class
        s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string

    # Write results
    for *xyxy, conf, cls in reversed(det):
        if save_txt:  # Write to file
            xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()  # normalized xywh
            line = (cls, *xywh, conf) if save_conf else (cls, *xywh)  # label format
            with open(f'{txt_path}.txt', 'a') as f:
                f.write(('%g ' * len(line)).rstrip() % line + '\n')

        if save_img or save_crop or view_img:  # Add bbox to image
            c = int(cls)  # integer class
            label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {conf:.2f}')
            annotator.box_label(xyxy, label, color=colors(c, True))
        if save_crop:
            save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] / f'{p.stem}.jpg', BGR=True)

# Stream results
im0 = annotator.result()
if view_img:
    if platform.system() == 'Linux' and p not in windows:
        windows.append(p)
        cv2.namedWindow(str(p), cv2.WINDOW_NORMAL | cv2.WINDOW_KEEPRATIO)  # allow window resize (Linux)
        cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
```

```
            cv2.imshow(str(p), im0)
            cv2.waitKey(1)  # 1 millisecond

        # Save results (image with detections)
        if save_img:
            if dataset.mode == 'image':
                cv2.imwrite(save_path, im0)
            else:  # 'video' or 'stream'
                if vid_path[i] != save_path:  # new video
                    vid_path[i] = save_path
                    if isinstance(vid_writer[i], cv2.VideoWriter):
                        vid_writer[i].release()  # release previous video writer
                    if vid_cap:  # video
                        fps = vid_cap.get(cv2.CAP_PROP_FPS)
                        w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
                        h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
                    else:  # stream
                        fps, w, h = 30, im0.shape[1], im0.shape[0]
                    save_path = str(Path(save_path).with_suffix('.mp4'))  # force *.mp4 suffix
on results videos
                    vid_writer[i] = cv2.VideoWriter(save_path,
cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
                vid_writer[i].write(im0)

    # Print time (inference-only)
    LOGGER.info(f"{s}{'' if len(det) else '(no detections), '}{dt[1].dt * 1E3:.1f}ms")

# Print results
t = tuple(x.t / seen * 1E3 for x in dt)  # speeds per image
```

```
    LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per image at shape {(1, 3, *imgsz)}' % t)
    if save_txt or save_img:
        s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}" if save_txt else ''
        LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")
    if update:
        strip_optimizer(weights[0])  # update model (to fix SourceChangeWarning)



def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'yolov5s.pt', help='model path or triton URL')
    parser.add_argument('--source', type=str, default=ROOT / 'data/images', help='file/dir/URL/glob/screen/0(webcam)')
    parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
    parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
    parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
    parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--view-img', action='store_true', help='show results')
    parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
    parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
    parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
```

```
    parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
    parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
    parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
    parser.add_argument('--augment', action='store_true', help='augmented inference')
    parser.add_argument('--visualize', action='store_true', help='visualize features')
    parser.add_argument('--update', action='store_true', help='update all models')
    parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
    parser.add_argument('--name', default='exp', help='save results to project/name')
    parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
    parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
    parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
    parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
    parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
    parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
    parser.add_argument('--vid-stride', type=int, default=1, help='video frame-rate stride')
    opt = parser.parse_args()
    opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1  # expand
    print_args(vars(opt))
    return opt


def main(opt):
    check_requirements(exclude=('tensorboard', 'thop'))
    run(**vars(opt))
```

```
if __name__ == "__main__":
    opt = parse_opt()
    main(opt)
```

Codes for the Livestream in Web Application

```
import cv2
from flask import Flask, render_template, Response
import torch
import numpy as np

app = Flask(__name__)

# Load custom YOLOv5 model from local path
model = torch.hub.load('ultralytics/yolov5', 'custom',
path='C:\\Users\\Shane\\Desktop\\webapp\\best.pt', force_reload=True)
model.eval()

# Custom label mapping
label_mapping = {
    0: 'SWAMP GAUGE',
    1: 'ONE FOOT',
    2: 'TWO FEET',
    3: 'THREE FEET',
    4: 'FOUR FEET',
    5: 'FIVE FEET',
    6: 'SIX FEET',
    # Add more labels as needed
}
```

```
# Function to start webcam and detect objects
def detect_objects():
    cap = cv2.VideoCapture(1)
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        # Perform object detection
        results = model(frame)

        # Get the bounding box coordinates and class labels
        boxes = results.xyxy[0].numpy()
        class_ids = results.pred[0].numpy()

        # Draw bounding boxes and labels on the frame
        for box, class_id in zip(boxes, class_ids):
            x1, y1, x2, y2, conf, cls = box
            label = label_mapping.get(int(cls), 'Unknown')  # Get the corresponding label or use 'Unknown' if not found
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (255, 0, 0), 2)
            cv2.putText(frame, label, (int(x1), int(y1) - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0), 2)

        # Encode the frame as JPEG
        _, img_encoded = cv2.imencode('.jpg', frame)

        # Yield the frame as a response
```

```python
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + img_encoded.tobytes() + b'\r\n')

# Route for accessing the webcam feed with object detection
@app.route('/video_feed')
def video_feed():
    return Response(detect_objects(), mimetype='multipart/x-mixed-replace; boundary=frame')

# Route for accessing the index page
@app.route('/')
def index():
    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Codes for the Main Page of the Website

(HTML)

```html
<!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" />
    <link rel="shortcut icon" type="x-icon" href="img/logo.png">
    <title>SWAMP</title>
```

```
    <!-- CSS -->
    <link rel="stylesheet" href="style.css" />

    <!-- Boxicons CSS -->
    <link href="https://unpkg.com/boxicons@2.1.2/css/boxicons.min.css"
rel="stylesheet"/>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@900&display=swap"
rel="stylesheet">
    <script src="https://kit.fontawesome.com/a076d05399.js"></script>
</head>

<body>
    <div class="banner">
        <div class="bar">
            <ul>
                <li><a href="faqs2">FAQs</a></li>
                <li><a href="aboutus2">About Us</a></li>
                <li><a href="contactus2">Contact Us</a></li>
            </ul>
        </div>
    </div>

    <div class="wrapper">
        <img class="logo" src="img/logo.png">
        <article class="info">
```

```
        <h1>Welcome to <span class="SWAMP">S.W.A.M.P</span></h1>
        <p>Monitor flood-prone areas in <span class="name">Brgy. Frances, Calumpit,
Bulacan </span>; be alerted and notified when the water level rises.</p>
        <div class = "login register">
          <a href="login"><button><span>LOGIN</span></button></a>
          <a href="register"><button><span>REGISTER</span></button></a>
        </div>
      </article>
    </div>

    <script>
      function showNotification() {
        const notification = new Notification("New message from SWAMP!", {
          body: "Hi, this is SWAMP."
        });

        notification.onclick = (e) => {
          window.location.href = "https://tupm-swamp.tk";
        }
      }

      console.log(Notification.permission);

      if (Notification.permission === "granted") {
        showNotification();
      } else if (Notification.permission !== "denied") {
        Notification.requestPermission().then(permission => {
          if (permission === "granted") {
            showNotification();
```

```
            }
        });
    }
  </script>
</body>
</html>
```

(CSS)

```
/* Google Fonts - Poppins */
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display
=swap");


* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
  text-decoration: none;
}
body {
  width: 100%;
  height: 100vh;
  background-image: linear-
gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(img/background.png);
  background-size: cover;
  background-position: center;
  background-attachment: fixed;
}
```

```
.bar{
    width: 100%;
    margin: auto;
    padding: 50px;
    padding-left: 64%;
    display: flex;
    justify-content: space-between;
    position: absolute;
    float: right;
}
.bar ul li{
    list-style: none;
    display: inline-block;
    margin: 0 30px;
    position: relative;
}
.bar ul li a{
    color: #fff;
    font-weight: 500;
    font-size: 18px;
}
.bar ul li::after{
    content: '';
    height: 3px;
    width: 0;
    background: #A3DF02;
    position: absolute;
    left: 0;
```

```
    bottom: -4px;
    transition: 0.5s;
}
.bar ul li:hover:after{
    width: 100%;
}
.wrapper{
    width: 100%;
    position: relative;
    top: 50%;
    transform: translateY(-50%);
    text-align: center;
    color: #fff;
    display: grid;
    grid-template-columns: 20% 80%;
    padding-left: 10%;
}
.info {
    width: 90%;
    float: right;
    padding-left: 100px;
    text-align: left;
    display: flex;
    flex-direction: column;
}
.info h1 {
    padding-top: 55px;
    font-size: 55px;
```

```
    color: #fff;
    line-height: 50px;
}
.info p {
    padding: 40px 0px 0px;
    font-family: inherit;
    font-size: 30px;
    color: #fff;
    line-height: 40px;
}
.name {
    color: #A3DF02;
}
.SWAMP {
    color: #A3DF02;
    font-family: 'Poppins', sans-serif;
}
.logo {
    width: 120%;
    padding-top: 40px;
}
.login {
    display: flex;
}
.register {
    justify-content: flex-start;
}
button {
```

```
    position: relative;
    margin-bottom: 25px;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 30px;
    height: 55px;
    width: 200px;
    background: #A3DF02;
    font-family: 'Poppins', sans-serif;
    box-shadow: 0px 6px 24px 0px rgba(0, 0, 0, 0.2);
    overflow: hidden;
    border: none;
    top: 40px;
    font-size: 15px;
    margin: 10px;
}
button:after {
    content: " ";
    width: 0%;
    height: 100%;
    background: #fff;
    position: absolute;
    transition: all 0.4s ease-in-out;
    right: 0;
}
button:hover::after {
    right: auto;
```

```
    left: 0;
    width: 100%;
}
button span {
    text-align: center;
    text-decoration: none;
    width: 100%;
    padding: 20px 25px;
    color: #111;
    font-size: 1.125em;
    font-weight: 600;
    letter-spacing: 0.2em;
    z-index: 20;
    transition: all 0.3s ease-in-out;
}
button:hover span {
    color: #A3DF02;
    animation: scaleUp 0.3s ease-in-out;
}
@keyframes scaleUp {
    0% {
    transform: scale(1);
    }
    50% {
    transform: scale(0.95);
    }
    100% {
    transform: scale(1);
```

```
    }
}

@media only screen and (max-width: 768px) {
    body {
        background-image: linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(img/background1.png);
      }
    .bar{
        width: 100%;
        margin: auto;
        padding: 30px;
        display: flex;
        justify-content: flex-end;
        position: absolute;
        float: right;
    }
    .bar ul li{
        margin: 0 10px;
    }
    .bar ul li a{
        font-size: 12px;
    }
    .wrapper{
        display: flex;
        flex-direction: column;
        padding-left: 0%;
    }
    .logo {
```

```
    width: 20%;
    margin: auto;
}
.info {
    width: 100%;
    padding: 0px 20px;
    text-align: center;
}
.info h1 {
    padding-top: 20px;
    font-size: 28px;
}
.info p {
    font-size: 14px;
    line-height: 20px;
    padding-top: 20px;
}
.login {
    display: flex;
    margin: auto;
}
button {
    height: 30px;
    width: 130px;
    top: 30px;
    font-size: 10px;
    margin: 10px;
}
```

ANNEX IV

DATA

Prediction Results

Date & Time	Actual Flood Level	Predicted Flood Level	Mean Absolute Percentage Error
6.17.2023 00:00:00	40	39.54	1.15%
6.17.2023 01:00:00	38	37.32	1.79%
6.17.2023 02:00:00	39	38.27	1.87%
6.17.2023 03:00:00	40	39.18	2.05%
6.17.2023 04:00:00	39	37.69	3.36%
6.17.2023 05:00:00	37	40.12	8.43%
6.17.2023 06:00:00	40	38.89	2.78%
6.17.2023 07:00:00	37	37.75	2.03%
6.17.2023 08:00:00	39	38.41	1.51%
6.17.2023 09:00:00	38	37.87	0.34%
6.17.2023 10:00:00	40	38.56	3.60%
6.17.2023 11:00:00	37	39.62	7.08%
6.17.2023 12:00:00	38	37.23	2.03%
6.17.2023 13:00:00	39	39.45	1.15%
6.17.2023 14:00:00	40	40.86	2.15%
6.17.2023 15:00:00	38	37.11	2.34%
6.17.2023 16:00:00	37	38.02	2.76%
6.17.2023 17:00:00	39	39.99	2.54%
6.17.2023 18:00:00	40	41.34	3.35%
6.17.2023 19:00:00	38	40.10	5.53%
6.17.2023 20:00:00	39	37.96	2.67%
6.17.2023 21:00:00	37	39.48	6.70%
6.17.2023 22:00:00	40	40.25	0.63%
6.17.2023 23:00:00	38	39.60	4.21%
6.18.2023 00:00:00	37	37.66	1.78%
6.18.2023 01:00:00	40	38.83	2.93%
6.18.2023 02:00:00	39	39.34	0.87%

6.18.2023 03:00:00	38	40.77	7.29%
6.18.2023 04:00:00	37	37.59	1.59%
6.18.2023 05:00:00	39	38.19	2.08%
6.18.2023 06:00:00	37	40.72	10.05%
6.18.2023 07:00:00	38	37.05	2.50%
6.18.2023 08:00:00	39	40.60	4.10%
6.18.2023 09:00:00	37	40.81	10.30%
6.18.2023 10:00:00	40	37.09	7.27%
6.18.2023 11:00:00	39	38.84	0.41%
6.18.2023 12:00:00	38	39.40	3.68%
6.18.2023 13:00:00	40	40.28	0.70%
6.18.2023 14:00:00	37	39.06	5.57%
6.18.2023 15:00:00	39	38.63	0.95%
6.18.2023 16:00:00	38	40.15	5.66%
6.18.2023 17:00:00	40	38.10	4.75%
6.18.2023 18:00:00	39	37.47	3.92%
6.18.2023 19:00:00	37	40.84	10.38%
6.18.2023 20:00:00	38	38.36	0.95%
6.18.2023 21:00:00	40	39.87	0.33%
6.18.2023 22:00:00	39	40.77	4.54%
6.18.2023 23:00:00	37	37.49	1.32%
6.19.2023 00:00:00	40	39.04	2.40%
6.19.2023 01:00:00	39	38.75	0.64%
6.19.2023 02:00:00	38	40.07	5.45%
6.19.2023 03:00:00	40	39.47	1.33%
6.19.2023 04:00:00	37	37.23	0.62%
6.19.2023 05:00:00	39	39.88	2.26%
6.19.2023 06:00:00	38	37.14	2.26%
6.19.2023 07:00:00	40	40.72	1.80%

6.19.2023 08:00:00	39	40.16	2.97%
6.19.2023 09:00:00	37	37.81	2.19%
6.19.2023 10:00:00	40	39.20	2.00%
6.19.2023 11:00:00	38	38.10	0.26%
6.19.2023 12:00:00	39	40.18	3.03%
6.19.2023 13:00:00	37	38.44	3.89%
6.19.2023 14:00:00	40	37.56	6.10%
6.19.2023 15:00:00	38	39.33	3.50%
6.19.2023 16:00:00	39	38.43	1.46%
6.19.2023 17:00:00	37	40.35	9.05%
6.19.2023 18:00:00	40	39.22	1.95%
6.19.2023 19:00:00	39	37.58	3.64%
6.19.2023 20:00:00	37	40.25	8.78%
6.19.2023 21:00:00	38	39.55	4.08%
6.19.2023 22:00:00	40	37.43	6.43%
6.19.2023 23:00:00	37	40.71	10.03%
6.20.2023 00:00:00	39	39.78	2.00%
6.20.2023 01:00:00	38	38.48	1.26%
6.20.2023 02:00:00	40	39.65	0.88%
6.20.2023 03:00:00	39	37.26	4.46%
6.20.2023 04:00:00	37	40.41	9.22%
6.20.2023 05:00:00	38	39.64	4.32%
6.20.2023 06:00:00	40	37.12	7.20%
6.20.2023 07:00:00	38	38.76	2.00%
6.20.2023 08:00:00	37	40.84	10.38%
6.20.2023 09:00:00	40	38.94	2.65%
6.20.2023 10:00:00	39	39.53	1.36%
6.20.2023 11:00:00	38	40.17	5.71%
6.20.2023 12:00:00	37	37.78	2.11%

6.20.2023 13:00:00	39	38.24	1.95%
6.20.2023 14:00:00	40	40.07	0.18%
6.20.2023 15:00:00	38	37.29	1.87%
6.20.2023 16:00:00	37	39.86	7.73%
6.20.2023 17:00:00	40	38.09	4.77%
6.20.2023 18:00:00	37	39.60	7.03%
6.20.2023 19:00:00	39	40.66	4.26%
6.20.2023 20:00:00	38	37.14	2.26%
6.20.2023 21:00:00	40	38.61	3.48%
6.20.2023 22:00:00	37	39.21	5.97%
6.20.2023 23:00:00	38	40.03	5.34%
6.21.2023 00:00:00	39	38.69	0.79%
6.21.2023 01:00:00	40	37.61	5.98%
6.21.2023 02:00:00	38	39.68	4.42%
6.21.2023 03:00:00	37	37.28	0.76%
6.21.2023 04:00:00	39	40.95	5.00%
6.21.2023 05:00:00	40	39.25	1.88%
6.21.2023 06:00:00	38	38.84	2.21%
6.21.2023 07:00:00	37	40.03	8.19%
6.21.2023 08:00:00	40	37.50	6.25%
6.21.2023 09:00:00	39	39.34	0.87%
6.21.2023 10:00:00	38	37.25	1.97%
6.21.2023 11:00:00	40	40.89	2.23%
6.21.2023 12:00:00	37	39.71	7.32%
6.21.2023 13:00:00	39	38.56	1.13%
6.21.2023 14:00:00	37	40.58	9.68%
6.21.2023 15:00:00	40	37.68	5.80%
6.21.2023 16:00:00	39	38.30	1.79%
6.21.2023 17:00:00	38	39.54	4.05%

6.21.2023 18:00:00	40	40.56	1.40%
6.21.2023 19:00:00	37	39.50	6.76%
6.21.2023 20:00:00	39	37.51	3.82%
6.21.2023 21:00:00	37	40.30	8.92%
6.21.2023 22:00:00	40	38.14	4.65%
6.21.2023 23:00:00	37	39.06	5.57%
6.22.2023 00:00:00	39	37.21	4.59%
6.22.2023 01:00:00	38	40.91	7.66%
6.22.2023 02:00:00	40	37.69	5.78%
6.22.2023 03:00:00	38	39.04	2.74%
6.22.2023 04:00:00	39	38.94	0.15%
6.22.2023 05:00:00	40	40.79	1.98%
6.22.2023 06:00:00	37	37.29	0.78%
6.22.2023 07:00:00	39	38.38	1.59%
6.22.2023 08:00:00	38	39.23	3.24%
6.22.2023 09:00:00	40	40.27	0.68%
6.22.2023 10:00:00	37	37.43	1.16%
6.22.2023 11:00:00	39	38.91	0.23%
6.22.2023 12:00:00	38	40.75	7.24%
6.22.2023 13:00:00	40	39.57	1.08%
6.22.2023 14:00:00	39	38.59	1.05%
6.22.2023 15:00:00	37	37.03	0.08%
6.22.2023 16:00:00	40	40.09	0.23%
6.22.2023 17:00:00	38	39.68	4.42%
6.22.2023 18:00:00	37	38.45	3.92%
6.22.2023 19:00:00	39	40.12	2.87%
6.22.2023 20:00:00	38	37.19	2.13%
6.22.2023 21:00:00	40	39.06	2.35%
6.22.2023 22:00:00	37	37.11	0.30%

6.22.2023 23:00:00	39	40.18	3.03%
6.23.2023 00:00:00	38	39.93	5.08%
6.23.2023 01:00:00	40	38.10	4.75%
6.23.2023 02:00:00	37	37.40	1.08%
6.23.2023 03:00:00	39	39.46	1.18%
6.23.2023 04:00:00	38	37.54	1.21%
6.23.2023 05:00:00	40	40.44	1.10%
6.23.2023 06:00:00	37	37.05	0.14%
6.23.2023 07:00:00	39	39.48	1.23%
6.23.2023 08:00:00	40	38.47	3.83%
6.23.2023 09:00:00	37	40.92	10.59%
6.23.2023 10:00:00	39	38.53	1.21%
6.23.2023 11:00:00	38	39.25	3.29%
6.23.2023 12:00:00	40	40.52	1.30%
6.23.2023 13:00:00	38	37.80	0.53%
6.23.2023 14:00:00	37	39.67	7.22%
6.23.2023 15:00:00	40	38.44	3.90%
6.23.2023 16:00:00	39	40.03	2.64%
6.23.2023 17:00:00	37	37.26	0.70%
6.23.2023 18:00:00	39	39.02	0.05%
6.23.2023 19:00:00	40	38.25	4.38%
6.23.2023 20:00:00	38	40.72	7.16%
6.23.2023 21:00:00	40	39.30	1.75%
6.23.2023 22:00:00	37	37.31	0.84%
6.23.2023 23:00:00	39	40.43	3.67%
6.24.2023 00:00:00	40	38.59	3.52%
6.24.2023 01:00:00	38	37.50	1.32%
6.24.2023 02:00:00	37	39.88	7.78%
6.24.2023 03:00:00	40	38.07	4.83%

6.24.2023 04:00:00	39	40.44	3.69%
6.24.2023 05:00:00	38	37.14	2.26%
6.24.2023 06:00:00	40	39.56	1.10%
6.24.2023 07:00:00	39	38.72	0.72%
6.24.2023 08:00:00	37	40.08	8.32%
6.24.2023 09:00:00	38	37.26	1.95%
6.24.2023 10:00:00	40	39.21	1.98%
6.24.2023 11:00:00	37	38.77	4.78%
6.24.2023 12:00:00	39	40.52	3.90%
6.24.2023 13:00:00	40	37.33	6.68%
6.24.2023 14:00:00	37	39.45	6.62%
6.24.2023 15:00:00	38	40.78	7.32%
6.24.2023 16:00:00	40	37.04	7.40%
6.24.2023 17:00:00	39	39.58	1.49%
6.24.2023 18:00:00	38	38.83	2.18%
6.24.2023 19:00:00	37	40.62	9.78%
6.24.2023 20:00:00	40	38.03	4.93%
6.24.2023 21:00:00	39	37.69	3.36%
6.24.2023 22:00:00	38	40.04	5.37%
6.24.2023 23:00:00	37	39.70	7.30%

Evaluation Results (Computed Mean Score)

Statements	**1**		**2**		**3**		**4**		**Total Score**	**Mean Score**	**Rating**
	x	**%**	**x**	**%**	**x**	**%**	**x**	**%**			
Functionality	3	3.75	5	6.25	21	26.25	52	65	284	3.55	Strongly Agree
Reliability	0	0	1	1.25	30	37.5	49	61.25	288	3.6	Strongly Agree
Usability	0	0	0	0	21	35	39	65	219	3.65	Strongly Agree
Efficiency	0	0	0	0	14	35	26	65	146	3.65	Strongly Agree
Maintainability	0	0	1	1.25	37	46.25	41	51.25	277	3.4625	Strongly Agree
Portability	0	0	6	10	20	33.33	34	56.67	208	3.4667	Strongly Agree
Overall Mean Score										**3.5632**	**Strongly Agree**

ANNEX V

SWAMP USER MANUAL

SWAMP:

Flood Monitoring, Prediction, and Alert System Using Wireless Sensor Network and Internet of Things (IOT)

User's Manual

OVERVIEW

SWAMP (System for Water Level and Alert Monitoring Pole) is a device designed to provide early detection of floods, alerting users up to 3 hours in advance of the event. It is accompanied by a user-friendly web application that displays real-time flood level data, along with other contributing parameters such as precipitation rate, air pressure, and water level. Additionally, the device includes a siren system that provides an additional notification when the flood level reaches a predefined threshold set by the water level scaler. With the SWAMP device and its accompanying web application, residents no longer need to rely on traditional flood detection methods, such as using a calendar or manually notifying individuals. This advanced system offers a more efficient and proactive approach to flood detection, enhancing the safety and preparedness of the community.

SWAMP
Ayala Blvd, Ermita,
Manila, 1000 Metro
Manila

SAFETY PRECAUTIONS

1. To ensure personal safety and prevent any potential harm, it is crucial to adhere to the provided instructions. Following these guidelines will significantly minimize the risk of injuries or accidents.

2. Electrical conduits serve as a protective barrier for wiring, shielding them from potential dangers such as heat and liquids. It is crucial to exercise caution when inspecting or repairing these wirings to ensure they are kept at a safe distance from water sources, as contact between live wires and water can lead to electrocution hazards. By adhering to these precautions, the risk of electrical accidents can be minimized, promoting safety in the maintenance and handling of electrical systems.

3. To prevent accidents, it is crucial to ensure the pole remains upright and stable. The main panel box, housing essential circuitry, must be handled with care and not subjected to dropping or impact, as it can result in system malfunctions. By prioritizing the pole's stability and protecting the main panel box and the sensors' platform, the risk of damage and subsequent operational issues can be minimized.

A. Setting Up the Device

Pole Structure

1. Gather all the necessary materials and tools, including the galvanized steel pole, solar panel, electric panel box, sensor platform, bolts, nuts, washers, and appropriate tools for installation.

2. Choose a suitable location for the pole, considering factors such as sunlight exposure for the solar panel and accessibility for maintenance and sensor readings.

3. Put the tire in a selected location, ensuring it to securely anchors the pole. The depth will depend on the soil conditions.

4. Place the galvanized steel pole into the tire, ensuring it is upright and straight. Use a level to verify its alignment.

5. Fill the tire with concrete or another suitable foundation material to secure the pole in place. Ensure that the pole remains straight during this process.

6. Allow the concrete to cure and harden.

7. Attach the solar panel to the top of the pole using the provided mounting brackets. Make sure the solar panel is facing in the correct direction to maximize sunlight exposure.

8. Connect the solar panel's wiring to the electric panel box. Take necessary precautions to ensure proper electrical connections and safety.

9. Install the delicate parts, such as microcontrollers and lead acid battery, inside the electric panel box. Ensure they are properly secured and protected from moisture.

10. Mount the sensor platform onto the pole at a suitable height, considering the specific requirements of the sensors. Use the provided mounting brackets or hardware to secure it in place.

11. Connect the sensors to the sensor platform, ensuring proper wiring and connections.

12. Test the entire system to verify that the solar panel, electric panel box, and sensors are functioning correctly. Check for any loose connections or potential issues.

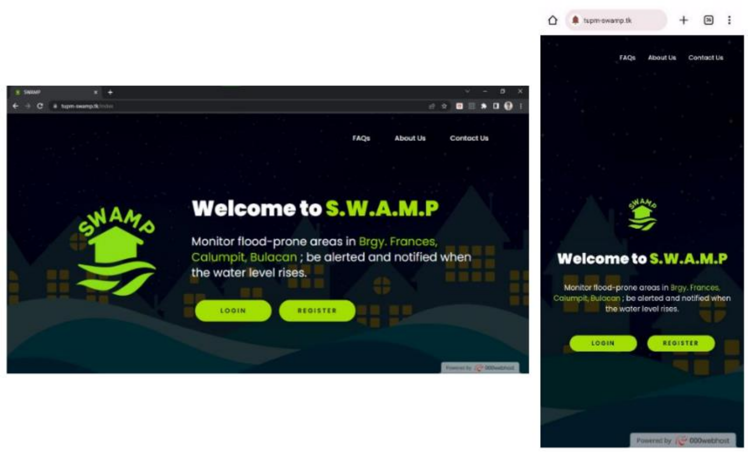
13. If necessary, adjust the positioning or alignment of the solar panel and sensor platform to optimize their performance.

14. Once everything is properly installed and tested, secure any loose wires or cables to the pole to ensure they are protected and won't interfere with the system's operation.

15. Regularly inspect and maintain the pole, solar panel, electric panel box, and sensor platform. This may include cleaning, checking connections, and monitoring sensor readings.

B. Using the Website Application

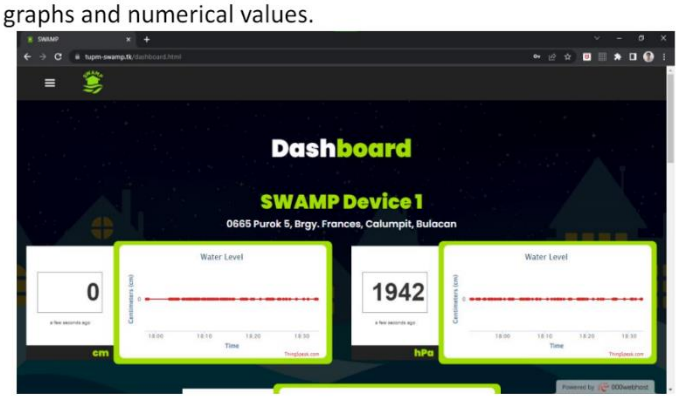
1. Open your preferred web browser and type in the link "***tupm-swamp.tk***" or you can just scan the QR code provided below to be redirected to the website. A welcome message will appear with two buttons: Register and Log In.



2. Signing Up/Register:
    - Type in your preferred Username, Name, and Password to create an account to be able to access the overall functions of the website.
    - Click on the "Sign Up" button and you will be redirected to the Dashboard Page.
    - If you already have an account, click on the "Log In" button.

3. Logging In:
    - Type in the Username and Password provided during your account creation process.
    - Click on the “Log In” button and you will be redirected to the Dashboard Page.
    - Make sure you have done the signing up/register process to be able to log in with your credentials.

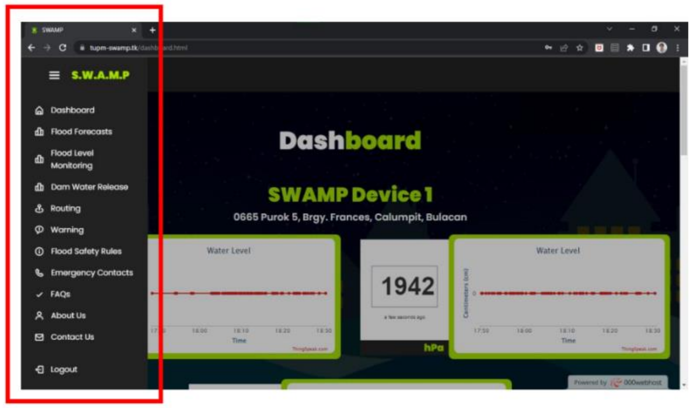
4. Dashboard Overview:
    - After signing up and logging in, you will be redirected to the website’s dashboard.
    - The dashboard contains the real time data from the sensors shown in graphs and numerical values.



    - On the dashboard, you will see a hamburger button (triple bar or three bold horizontal lines resembling a hamburger) located on the top left side corner of the website.

5. Hamburger Button:
    - This button contains the other features of the website; Flood Forecasts, Flood Level Monitoring, Dam Water Release, Routing,

Warning, Flood Safety Rules, Emergency Contacts, Frequently Asked Questions (FAQs), About Us, Contact Us, and Logout button.



- Click on your desired feature of information and you will be redirected to it.

6. Flood Forecasts Overview:
    - The flood forecasts page contains information such as the predicted flood level shown on a graph, an alert message, and the real time data of the rainfall and water level based on the data acquired from the sensors. The average basin rainfall and water level are based according to the category shown on the website.

7. Flood Level Monitoring Overview:
    - This contains the live feed of the camera for monitoring the flood level and a water level height category that is the same on the meter scale shown on the live feed. The water level height is compared to the height of an average Filipino man to be able to know how high the flood is.
8. Dam Water Release Overview:
    - This contains a real time update of the reservoir water level of each dam. The data shown is from the website of the PAGASA.

9. Routing Overview:
    - This contains a map of the Barangay Frances in Calumpit, Bulacan. It has pinned points of the three (3) nodes of our device, the barangay hall, and the evacuation area.

10. Flood Safety Rules Overview:
    - This contains the safety rules to be followed before, when warned, during, and after a flood.
    - It also has the things one can do to mitigate floods.
    - These safety rules are copied from the PAGASA website.

11. Emergency Contacts Overview:
    - This contains the emergency contacts that can be called and texted upon the occurrence of a flood.
    - It has several individuals such as the barangay officials and the local government units.

12. Frequently Asked Questions (FAQs) Overview:
    - FAQs answer the basic questions of a user upon using this website.

13. About Us Overview:
    - The about us page contains a brief overview of us researchers and our filled roles in conducting this project.

14. Contact Us Overview:
    - This page can be used when you have a question about the device and for us to respond immediately upon asked. This can also be used to report a problem regarding our device.
    - It contains our location, contact number, and email account.

15. Logout:
    - The logout button when clicked will automatically redirect you to the main page of the website.

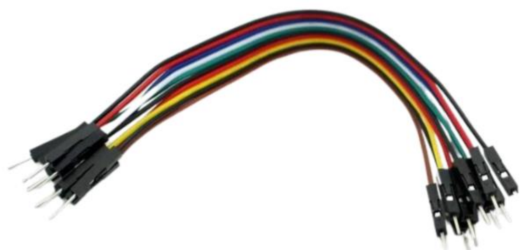
C. Troubleshooting/Maintenance

Note: This section of the user manual is only applicable to authorized persons who have access to the component box of the system.

In case of any problems, here are the steps to follow in order to ensure that the system is working properly:

1. Check the solar charge controller. If the icon shows that the battery percentage is low, it might be an indication that the battery is not charging properly. Ensure that the royal cables are connected properly. The **red** wire should be connected to the **positive** terminal, while the **black** wire should be connected to the **negative** terminal.

2. Ensure that the wires of sensors are properly connected to the Nodemcu microcontroller. The jumper wires can be easily connected by inserting each wire to its corresponding color.



3. Inspect the catch basin of the rain gauge. Remove the accumulated dirt, leaves, or other materials inside the tipping bucket rain gauge.

4. Make sure that the water level scaler is free from any obstruction from the camera.

5. If problems continue to occur, try rebooting the power using the USB hub.

ANNEX VI

MANUAL FOR DUPLICATION OF PROTOTYPE

A MANUAL FOR DUPLICATING THE PROTOTYPE

A. HARDWARE

a. Pole Structure

1. Gather all the necessary materials and tools, including the galvanized steel pole, solar panel, electric panel box, sensor platform, bolts, nuts, washers, and appropriate tools for installation.
2. Choose a suitable location for the pole, considering factors such as sunlight exposure for the solar panel and accessibility for maintenance and sensor readings.
3. Put the tire selected location, ensuring it to securely anchor the pole. The depth will depend on the soil conditions.
4. Place the galvanized steel pole into the tire, ensuring it is upright and straight. Use a level to verify its alignment.
5. Fill the tire with concrete or another suitable foundation material to secure the pole in place. Ensure that the pole remains straight during this process.
6. Allow the concrete to cure and harden.
7. Attach the solar panel to the top of the pole using the provided mounting brackets. Make sure the solar panel is facing in the correct direction to maximize sunlight exposure.
8. Connect the solar panel's wiring to the electric panel box. Take necessary precautions to ensure proper electrical connections and safety.

9. Install the delicate parts, such as microcontrollers and lead acid battery, inside the electric panel box. Ensure they are properly secured and protected from moisture.
10. Mount the sensor platform onto the pole at a suitable height, considering the specific requirements of the sensors. Use the provided mounting brackets or hardware to secure it in place.
11. Connect the sensors to the sensor platform, ensuring proper wiring and connections.
12. Test the entire system to verify that the solar panel, electric panel box, and sensors are functioning correctly. Check for any loose connections or potential issues.
13. If necessary, adjust the positioning or alignment of the solar panel and sensor platform to optimize their performance.
14. Once everything is properly installed and tested, secure any loose wires or cables to the pole to ensure they are protected and won't interfere with the system's operation.
15. Regularly inspect and maintain the pole, solar panel, electric panel box, and sensor platform. This may include cleaning, checking connections, and monitoring sensor readings.

b. Water Level Scaler

Materials needed:

- Steel scale divided into 6 feet sections

- Yellow paint
- Orange paint
- Red paint
- Paintbrush
- Labels

1. Prepare the steel scale. Ensure that the steel scale is clean and free from any debris or dirt. If necessary, use a damp cloth to wipe off any dust or dirt.
2. Divide the steel scale into 6 feet sections. Measure and mark equal intervals on the steel scale, starting from the bottom. Use a ruler or measuring tape to ensure accuracy. Divide the scale into six 1-foot sections, labelling them from 1 to 6 feet.
3. Prepare the paint colours. Open the yellow paint and stir it well to ensure it is thoroughly mixed. Pour enough yellow paint into a container or palette. Repeat the process for the orange and red paints.
4. Paint the first 2 feet section with yellow. Dip the paintbrush into the yellow paint. Apply the yellow paint evenly on the first 2 feet section of the steel scale. Allow the paint to dry completely before moving on to the next step.
5. Paint the next 2 feet section with orange. Clean the paintbrush or use a new one for the orange paint. Apply the orange paint evenly on the next 2 feet section of the steel scale. Ensure a clear separation between the yellow and orange sections. Allow the paint to dry completely.

6. Paint the remaining 2 feet section with red. Clean the paintbrush or use a new one for the red paint. Apply the red paint evenly on the remaining 2 feet section of the steel scale. Ensure a clear separation between the orange and red sections. Allow the paint to dry completely.
7. Label the sections. Prepare labels with the number 1 to 6 feet.
8. Display and use the color-coded steel scale. Find a suitable location to display the scale where it is easily visible and accessible during flooding situations. Mount the scale securely on a wall or a stand. Ensure that the labels and painted sections are clearly visible. Educate relevant personnel or residents about the scale and its meaning:

   Yellow section (1-2 feet) indicates the "Monitor" level, which suggests heightened awareness and preparedness for potential flooding.

   Orange section (3-4 feet) represents the "Alert" level, indicating that flooding is likely or imminent.

   Red section (5-6 feet) signifies the "Evacuate" level, which means immediate evacuation is necessary due to severe flooding.

c. Sensors

1. Select a sensor that is appropriate for the weather conditions at the intended deployment site.
2. Calibrate the ultrasonic sensor. Consider the height of the sensor above ground level. Determine the formula depending on the temperature. The speed of sound

varies depending on the temperature. Measure the water level manually, then compare it to the reading of the sensor to determine its accuracy.

3. The barometric pressure sensor should not be placed in a closed case because it will obstruct the air flow. Make sure that the length of the royal cable connected to the sensor does not exceed 1 meter because it will affect the i2c terminal.
4. Calibrate the rain gauge. Replace the reed switch with a hall effect sensor. This sensor has more advantages compared to a reed switch because it is easier to configure and has a digital input that can be easily programmed. The formula for measuring the rainfall height is equal to the volume of rain collected over the catchment area. Likewise, the amount of rainfall can also be quantified by determining how much liquid has passed through the bucket per tip. After calibrating, the sensor measures 5 ml of liquid per tip.
5. Upload all the codes to the Nodemcu microcontroller. Interface it with ThingSpeak and then display the collected data to the official website.

B. SOFTWARE

a. Machine Learning for Prediction

Materials Needed:

- Computer
- Visual Studio Code
- Python
- Google Cloud Platform

STEP 1: Creating the Predictive Algorithm Program

1. Select the most suitable machine learning algorithm based on the nature of the dataset (e.g., linear regression, random forest, neural networks).
2. Prepare the dataset by ensuring it is properly formatted and contains the necessary features and target variables.
3. Split the dataset into training and testing sets using a suitable ratio (e.g., 80% for training and 20% for testing).
4. Normalize or scale the features in the dataset if required, using techniques such as min-max scaling or standardization.
5. Initialize and configure the chosen machine learning algorithm, setting the desired parameters and hyperparameters.
6. Train the algorithm using the training dataset, allowing it to learn patterns and relationships between the features and target variable.
7. Evaluate the accuracy of the trained model by predicting the target variable on the testing dataset and comparing the predictions with the actual values.
8. Fine-tune the model, if necessary, by adjusting the algorithm's parameters or exploring different algorithms.

STEP 2: Creating the Continuous Prediction Program

1. Load the trained model obtained from STEP 1 into the program.
2. Implement a continuous prediction functionality that takes input data, preprocesses it if required, and feeds it into the model for prediction.

3. Design a user-friendly output format that includes the predicted values, corresponding timestamps, and any additional relevant information.
4. Generate graphs or visualizations based on the predicted values to provide a clear representation of the trends and patterns.
5. Integrate the program with the preferred platform or system for seamless output delivery.
6. Test the continuous prediction program by feeding it with new data and monitoring the accuracy and reliability of the predictions.

STEP 3: Creating the Decision Classifier

1. Define the classes and critical levels that is necessary for the decision classifier to identify based on the predicted values.
2. Determine the appropriate decision-making criteria for classifying predictions into different categories (e.g., thresholds, rule-based logic).
3. Develop the decision classifier program that takes the predicted values as input and assigns them to the corresponding classes or critical levels.
4. Display the decision classifier's output on the preferred platform, ensuring it provides clear and easily interpretable information about the identified classes or critical levels.
5. Test the decision classifier by providing it with various predicted values and verifying that it accurately classifies them according to the predefined criteria.
6. Continuously monitor the performance of the decision classifier and update the criteria or rules as needed to improve its accuracy and relevance.

STEP 4: Ensuring Endless Program Execution

1. Implement a continuous looping mechanism in the programs to ensure they run endlessly.
2. Monitor the execution of the programs to ensure they are functioning as expected without any errors or interruptions.
3. Define conditions or triggers for human intervention, such as specific events or thresholds, to stop the programs when necessary.
4. Regularly review and maintain the programs, updating them as needed to incorporate improvements, address any issues, or accommodate changing requirements.

b. Image Processing for Monitoring

The following steps state the duplication process of the flood monitoring aspect of the system using the YOLOv5 model and a webcam:

1. Download the needed repositories of the YOLOv5 model from Ultralytics
2. Create the custom dataset needed for the labelling and weights of the model; the dataset must come from the water level gauge used by the camera
3. Train the YOLOv5 model; after training the model, it will extract the needed weights to use in testing the model in images, videos, and direct capturing from a webcam

4. Testing the model using the weights with the specific configuration from the Ultralytics module will display the validity of the accuracy of the model from the trained model
5. After getting the best accuracy of the model, create a web application that embeds the YOLOv5 model into the webcam while live streaming it
6. Finally, integrate the web application with the chosen cloud platform to display the monitoring aspect on a public web application

c. Website

1. Have a basic knowledge on how to use and code an HTML, CSS, JavaScript, and PHP language. These languages will build entirely the website and will be the highlight on the duplication of the website.
2. Choose a domain name that is unique and will best represent your website. We used "***tupm-swamp.tk***" and was purchased free from Freenom.
3. Select a web hosting provider that suits your needs like 000webhost and other free web hosting provider available online. This will depend on what your website needs in terms of pricing, server performance, storage, bandwidth, and customer support options.
4. To be able to change the link of the website and disregard the url of the web host used, changes will be done inside Freenom. Linking the main url of the website from 000webhost to Freenom and changing it to the desired url name will be done.

5. Incorporate Cloudflare which is free to improve the security, performance, and reliability of the website. To connect this to your DNS, it just needs to be pointed directly to the url of the main website. Prior changes will be done like adding a record of CNAME with the website domain name, content, proxy status, and TTL.
6. Customize first the website's appearance before uploading it to a web host. Select a theme/template that aligns with the website's purpose and customize its design. Modify colors, fonts, layouts, and add logo and branding elements to create a unique look. This will be programmed and coded using HTML, CSS, JavaScript, and PHP language.
7. Test the website before launching. Xampp can be used to be able to view the website on a localhost server. Check for any broken links, formatting issues, or functionality problems. Make sure the website is responsive and user-friendly across various web service platforms/browsers.
8. Once satisfied with the testing results, it's time to make the website live and available online. Point your domain to the web hosting server and ensure all files are uploaded correctly. Double-check that everything is working as intended.
9. Regularly update the website's content and address security updates. Also monitor the performance of the website, track analytics, and engage with the visitors and users of the website.

ANNEX VII

PROJECT DOCUMENTATION

Needs Assessment in Barangay Frances

Barangay Visits and Deployment Inspections

Project Proposal with Sangguniang Barangay

Project Deployment

Data Gathering

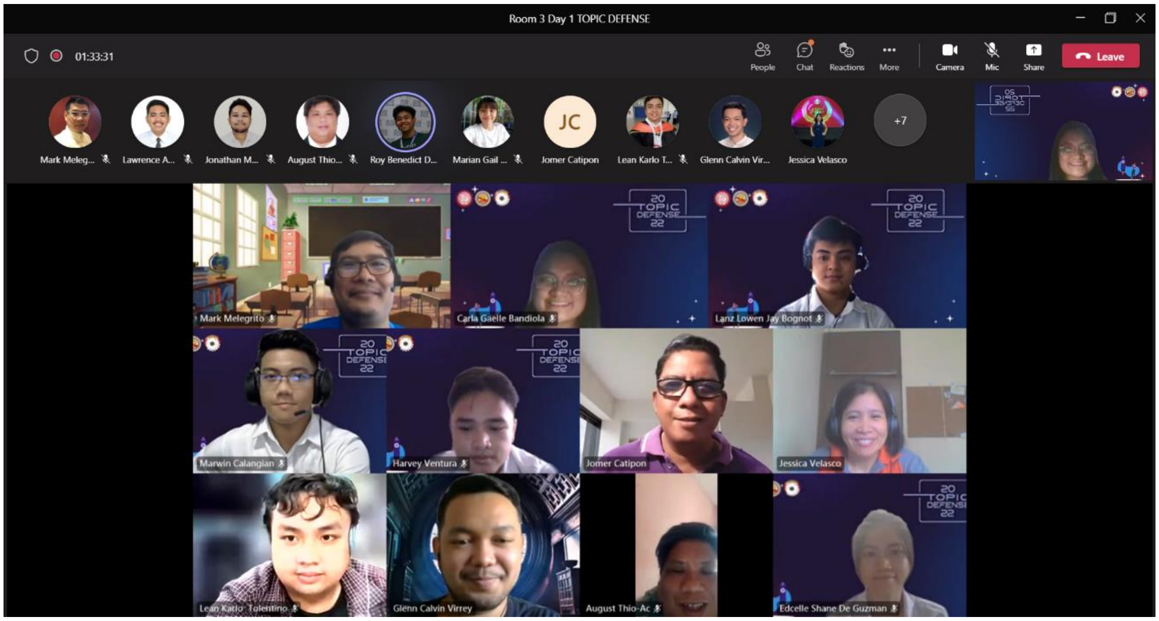
Deployment Checkups

Meeting with the Chief of PRFFWC-PAGASA

APPRECIATE 2023


SEEFLOODS
CK 'N FLARE
CAMERA

Project Defenses




GUZMAN

ANNEX VIII

PROPONENTS' PROFILE

CARLA GAELLE D. BANDIOLA

+639761533682
Bacoor City, Cavite
carlagaellebandiola@gmail.com

EDUCATION

BS IN Electronics Engineering
Technological University of the Philippines - Manila
2019-2023

Senior High School
University of Perpetual Help System DALTA - Las Piñas
2017 - 2019

Junior High School
St. Michael's Institute - Bacoor
2013 - 2017

SKILLS

- Python Programming
- Machine Learning
- NI Multisim
- PCB Designing
- Cisco Packet Tracer
- 3D Modeling
- Adobe Photoshop
- Data Encoding/Organization
- Research and Development
- Microsoft Applications
- Google Applications
- Project Management

ABOUT ME

Skilled electronics engineering graduate specialized in information, communications, and technology (ICT). Proficient in designing and implementing innovative electronic systems, with a solid understanding of telecommunications and network protocols. Proactive problem solver with strong analytical skills, dedicated to staying updated with emerging technologies and delivering successful projects within deadlines.

WORK EXPERIENCE

Research Development & Project Management Intern — 2022
Melham Construction Corporation I Quezon City

- Co-leader of the 11th ECE Pitch Team
- Managed daily meetings and documentation of tasks
- Developed a project study entitled "Design and Fabrication of Paper-based Printed Circuit Board (PCB) Using Water Hyacinths"

Freelance Graphic Designer & Content Writer — 2022
Bacoor City, Cavite

- Created and edited graphic designs, images, and visual contents
- Created and managed documents for clients
- Designed public materials such as election flyers, invitations, and brochures

Ecommerce Merchant — 2021
Facebook Marketplace

- sold personal items, food products, and household items via online transactions

REFERENCES

Engr. Mark P. Melegrito
Professor / Technological University of the Philippines - Manila
Email : mark_melegrito@tup.edu.ph

LANGUAGES

- ENGLISH
- FILIPINO

JONATHAN MICHAEL A. MANALO

+639455584371
Dasmariñas, Cavite
jonathanmichaelmanalo@gmail.com

EDUCATION

BS in Electronics Engineering
Technological University of the Philippines - Manila
2019 - 2023

Senior High School
Lyceum of the Philippines University - Cavite
2017 - 2019

Junior High School
Biga Achievers Learning Institute, Inc.
2013 - 2017

SKILLS

- Python Programming
- Machine Learning
- NI Multisim
- PCB Designing
- Cisco Packet Tracer
- Adobe After Effects
- Adobe Premiere Pro
- Adobe Photoshop
- Research and Development
- Microsoft Applications
- Google Applications

ABOUT ME

I am an Electronics Engineering graduate with a specialized focus on Information and Communications Technology (ICT). Equipped with proficient programming and editing skills, I am driven by a relentless pursuit of innovation and self-improvement. My leadership acumen enables me to foster effective collaboration and guide teams towards achieving shared objectives. Committed to staying at the forefront of technological advancements, I actively seek opportunities to broaden my expertise and make a substantial impact in the realm of technology.

WORK EXPERIENCE

Research Development & Project Management Intern 2022
Melham Construction Corporation I Quezon City

- Leader of the 11th ECE Pitch Team
- Developed a Project Study about Water Hyacinth Based PCB Manufacturing
- Developed a Project Study about Sign Language to Speech Translation

Freelance Graphic Designer 2020
Dasmariñas, Cavite

- Created modern food menu designs
- Created recipe visuals for a website
- Created public materials for small university events
- Edited videos for students, debuts, and weddings
- Edited Posters, Tarpaulin Designs, and Flyers for special events

Part-Time Musician 2018
Lyceum of the Philippines University I General Trias, Cavite

- Performed as a Lead Guitarist at various religious concerts
- Performed as a Lead Guitarist and Back-up Singer for debuts
- Performed as a Lead Guitarist and Back-up Singer for weddings
- Performed as a Lead Guitarist and Back-up Singer for public gigs

REFERENCES

Engr. Mark P. Melegrito
Professor / Technological University of the Philippines - Manila
Email : mark_melegrito@tup.edu.ph

LANGUAGES

- ENGLISH
- FILIPINO

HARVEY C. VENTURA

+639087625734
Bacoor City, Cavite
venturaharv@gmail.com

EDUCATION

BS in Electronics Engineering
Technological University of the Philippines - Manila
2019 - 2023

Senior High School
Theresian School of Cavite

2017 - 2019

Junior High School
St. Michael's Institute

2013 - 2017

SKILLS

- Arduino Programming
- Ni Multisim
- Proteus
- Cisco Packet Tracer
- GNS3
- Matlab/Octave
- Research and Development
- Microsoft Applications
- Google Applications

ABOUT ME

A hardworking individual seeking a full-time position where I can apply my knowledge and skills related to my undergraduate studies. I aim to continuously develop my technical skills through experience to be able to contribute to the company's vision.

WORK EXPERIENCE

Research Development & Project Management Intern 2022
Melham Construction Corporation I Quezon City

- Developed a Project Study about Water Hyacinth Based PCB Manufacturing
- Developed a Project Study about Sign Language to Speech Translation

REFERENCES

Engr. Mark P. Melegrito
Professor / Technological University of the Philippines - Manila
Email : mark_melegrito@tup.edu.pj

LANGUAGES

- **ENGLISH**
- **FILIPINO**

MARWIN R. CALANGIAN

+639288948314 Quezon City, Metro Manila

mcalangian0614@gmail.com

EDUCATION

BS in Electronics Engineering
Technological University of the Philippines - Manila
2019 - 2023

Senior High School
Polytechnic University of the Philippines - Manila
2017 - 2019

Junior High School
Batasan Hills National High School
2013 - 2017

SKILLS

- Python Programming
- C Programming
- Machine Learning
- NI Multisim
- PCB Designing
- Cisco Packet Tracer
- Research and Development
- Microsoft Applications
- Google Applications
- Research and Development
- Microsoft Applications
- Google Applications
- Web Development

ABOUT ME

I am an Electronics Engineering graduate with a passion for innovation and problem-solving. Skilled in emerging technologies. I thrive in designing and implementing technological solutions. With expertise in programming languages like C, and Python, I can bridge the gap between software and hardware. Collaborative and communicative, I enjoy working in interdisciplinary teams to achieve common goals. Continuous learning drives me to explore emerging technologies such as IoT and machine learning. I am ready to apply my skills and contribute to projects that make a meaningful impact on technology.

WORK EXPERIENCE

Quality Assurance Trainee 2022
CLIXLogic Inc. I Quezon City

- Assist in conducting quality inspections and tests on products/devices.
- Follow predefined quality control procedures and work instructions.
- Prepare reports summarizing the monitored data and for review by senior team members.

PUP-CS Laboratory Inventory Clerk 2019
Polytechnic University of the Philippines I Manila

- Maintain accurate records of laboratory inventory, including supplies, equipment, and chemicals.
- Maintain a schedule for equipment maintenance, calibration, and servicing.
- Ensure proper storage, labeling, and disposal of hazardous substances in accordance with regulations.
- Assist in training laboratory personnel on inventory management procedures and best practices.

Freelance Gadget Repair Technician 2018
Quezon City, Metro Manila

- Utilize diagnostic tools and techniques to troubleshoot and isolate hardware or software problems.
- Conduct software updates, installations, and configurations to resolve software-related issues.
- Perform repairs on a variety of electronic devices, such as smartphones, tablets, laptops, gaming consoles, and other consumer gadgets.

REFERENCES

Engr. Mark P. Melegrito
Professor / Technological University of the Philippines - Manila
Email : mark_melegrito@tup.edu.ph

LANGUAGES

- **ENGLISH**
- **FILIPINO**

LANZ LOWEN JAY O. BOGNOT

+639295459221
Gracepark, Caloocan City
lanzlowenjay.bognot@tup.edu.ph

EDUCATION

BS in Electronics Engineering
Technological University of the Philippines - Manila
2019 - 2023

Senior High School
Catanduanes State University Laboratory Schools
2017 - 2019

Junior High School
Viga Rural Development High School
2013 - 2017

SKILLS

- HTML
- CSS
- JavaScript
- PHP
- MS Dos
- Python Programming
- MATLAB/Octave
- NI Multisim
- Proteus, EKTS
- Cisco Packet Tracer
- GNS3
- Microsoft Applications
- Google Applications

ABOUT ME

I am determined to secure a cooperative education opportunity in the captivating realm of Electronics Engineering, as it promises to provide an invigorating challenge and fortify my academic and professional competencies. My aim is to embark on this journey with unwavering passion, surpassing the standards of excellence to continually enhance my abilities and skills, thereby fostering my personal growth in this field. I am committed to toiling diligently, striving for triumph in a stimulating and demanding environment, and making valuable contributions to the company's success while embracing the prospects for advancement that lie ahead. Armed with the knowledge I have inherited and the intelligence I have acquired, I am determined to surpass expectations through hard work, perseverance, and an unwavering dedication to excellence.

WORK EXPERIENCE

Project Management & Engineering Intern 2022
Sysnet Integrators, Inc. | Makati City

- Monitor RFID System through RFID reader logs
- Verify RFID response of the system through CCTV and database
- Design a process flow for a company project about EVCS (Electric Vehicle Charging Stations)
- Compare the flowchart's accuracy to the developed user website
- Test of the Electric Vehicle Charger
- Identify errors for every user transaction of the EVC
- Verify the EVC display's accuracy using the user's SMS display
- Prepare a list of feedback and suggestions to the web developer
- Assemble electronic components for the EVSE project for scooter

REFERENCES

Engr. Mark P. Melegrito
Faculty, ECE Department, Technological University of the Pilippines - Manila
CONTACT UPON REQUEST

Engr. Glenn C. Virrey, ECT
Faculty, ECE Department
University of Santo Tomas
CONTACT UPON REQUEST

LANGUAGES

- ENGLISH
- FILIPINO

EDCELLE SHANE D. DE GUZMAN

+639471083287

Sta. Ana, Manila

edcelleshane.deguzman@tup.edu.ph

EDUCATION

BS in Electronics Engineering
Technological University of the Philippines - Manila
2019 - 2023

Senior High School
Polytechnic University of the Philippines - Sta. Mesa
2017 - 2019

Junior High School
St. Paul College of San Rafael
2013 - 2017

SKILLS

- Python Programming
- Machine Learning
- MATLAB/Octave
- NI Multisim
- Cisco Packet Tracer
- Adobe Illustrator
- Adobe Premiere Pro
- Adobe Photoshop
- Sony Vegas Pro
- Research and Development
- Microsoft Applications
- Google Applications

ABOUT ME

A person who pushes for science and excellence to produce original and creative ideas that support the company's operations, the people, and the environment. I am currently searching for a full-time position in a company that adopts these abilities.

WORK EXPERIENCE

Software Development Intern 2022
Erovoutika I Santa Rosa City Laguna

- Lead the Intern Team
- Created a Prediction Application
- Established a Research Paper about Machine Learning

Graphic Designer 2022
Erovoutika I Santa Rosa City Laguna

- Designs Publication Materials
- Produces Webinar Materials

Office Assistant 2019
Office of the Vice President of Academic Affairs I PUP - Sta. Mesa

- Assists Students' Concern
- Arranges Office Papers
- Distributes Official Documents
- Receives Office Calls
- Inspects Office Attendance Forms

REFERENCES

Engr. Mark P. Melegrito
Faculty, ECE Department Technological University of the Philippines - Manila
CONTACT UPON REQUEST

Engr. Glenn C. Virrey, ECT
Faculty, ECE Department
University of Santo Tomas
CONTACT UPON REQUEST

LANGUAGES

- ENGLISH
- FILIPINO