``

CS5053NP

Module Code & Module Title

Level 5 – Cloud computing and Internet of Things CS5053NP

Assessment Type

50% Group Coursework 1

Semester- 1

2022/23 Autumn

Assignment Due Date: Friday, January 12, 2024

Assignment Submission Date: Friday, January 12, 2024

Submitted To: Mr. Sandip Adhikari

Word Count (Where Required): 6678

Student Name	London Met ID	College ID
Prabal Gurung	22069041	NP04CP4A220088
Aman Gurung	22068950	NP04CP4A220019
Prabin Thapa	22069043	NP04CP4A220090
Rachit Raj Shrestha	22069056	NP04CP4A220101
Rahul G.C	22069057	NP04CP4A220102

Declaration

I confirm that I understand my coursework needs to be submitted online via My second teacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submission will be treated as non-submission and a mark of zero will be awarded

Abstract:

The project primary objective is to develop a comprehensive disaster management system integrated with a smart dam infrastructure.

This system combines real-time monitoring, predictive analytics, and IoT connectivity to enhance resilience against natural disasters. Additionally, the project aims to optimize water storage and release through an intelligent dam mechanism while ensuring agriculture irrigation. With community support, Aqua Sentinel will be able to make timely information and education initiative that alerts community before any disaster occurs.

Table of Contents:

1. Introduction: 1

1.1 Current Scenario: 2

1.2 Problem statement and project as solution: 6

1.3 Aim and Objectives: 7

2. Background: 8

2.1 System Overview: 8

2.2 Design Diagrams: 10

2.3 Requirement Analysis: 15

3. Development: 26

4. Results and findings: 42

4.1 Project Outcome: 42

4.2 Test cases: 43

5. Future works: 56

6. Conclusion: 57

7. References: 58

8. Appendix: 59

8.1 Source Code: 59

8.2 Pictures of the system: 81

8.3 Individual contribution plan: 82

Table of Figures:

Figure 1: Destruction caused by floodwaters in Melamchi bazaar. 3
Figure 2: Water overflowing after the diversion channel is blocked at the Arun III dam site on Sunday. 4
Figure 3: Model Diagram of our project 11
Figure 4: Circuit Diagram of our project 12
Figure 5: Flowchart of our project 13
Figure 6: System Architecture of our project 14
Figure 7: Arduino Uno 15
Figure 8: Jumper Wire 16
Figure 9: Ultrasonic Sensor 16
Figure 10: Soil Moisture Sensor 17
Figure 11: Buzzer 17
Figure 12: LED 18
Figure 13: Servo Motor 19
Figure 14: Bread Board 20
Figure 15: Water Pump Motor. 20
Figure 16: USB Cable 21
Figure 17: LCD 21
Figure 18: ESP32-CAM 22
Figure 19: Relay Module 22
Figure 20: Resistor 23
Figure 21: Microsoft Word 24
Figure 22: Vectary 24
Figure 23: Arduino IDE 25
Figure 24: Fritzing 25
Figure 25: Research of IOT system. 26
Figure 26: Requirement analysis by group member. 27
Figure 27: Collection of resources. 29
Figure 28: Testing of single component at a time. 31
Figure 29: Reference for design of system. 32

Figure 30: Designing circuit diagram in fritzing. 33
Figure 31: Development of system. 34
Figure 32 : Development of system II 35
Figure 33: Review and update of system 37
Figure 34 : Review and update of system II 38
Figure 35: Integration of system. 40
Figure 36: Testing of whole system. 41
Figure 37: Completed Project. 42
Figure 38: Testing for water pump. 43
Figure 39: Test for ultrasonic sensor 45
Figure 40: Test for valve of dam. 46
Figure 41: Test for alarm system. 48
Figure 42: Test for bridge. 50
Figure 43: Test for soil moisture. 52
Figure 44: Test for water pump to agricultural plot. 53
Figure 45: Test for guest perspective of website. 55
Figure 46: Completed picture of our project. 81
Figure 47: Work break down structure of project. 83

Table of Table:

Table 1: Test case no.1 44
Table 2: Test case no.2 45
Table 3:Test case no.3 47
Table 4: Test case no.4 49
Table 5: Test case no.5 51
Table 6: Test case no.6 52
Table 7: Test case no.7 54
Table 8: Test case no.8 55
Table 9: Individual Contribution. 82

1. Introduction:

In the face of escalating challenges posed by climate change, growing population demands and sustainable resource management, the Aqua Sentinel Smart Dam System emerges as a bacon of innovation. This project envisions a dynamic water infrastructure that not only stores and release water intelligently but also proactively engages with the needs of agriculture while safeguarding against potential crises.

Aqua Sentinel's core features include an intricate network of sensors, automated bridge management system and alarm that alerts civilization in case of emergency. It combines these elements to ensure optimal water storage, controlled release to prevent overflow, precision irrigation for agriculture as well as safeguard of infrastructure like bridges. The result of this project will harmonize human needs with ecological balance. This project aims to make an adaptive dam system that adapts to any environmental condition and promotes efficient uses of water in agriculture contexts. Beyond the conventional roles of dam, envisioning them not just as reservoirs but as intelligent hubs capable of making informed decisions in real-time. Such conversion of dams into intelligent hubs helps human in various ways. The settlement areas near the dam are saved from the coming crisis due to the ability of the dam to issue a warning to the people before the overflow of water in the reservoirs. Similarly, the automated bridge management system safeguards the bridge from getting demolished during the release of the water from the reservoir. In a world full of uncertainties, climate change is one of the dangerous aspects for a transitional village and Aqua Sentinel hopes to mitigate the damage in event of crisis.

As we go deeper in depths of Aqua Sentinel, this report will unveil the intricate design, technological architecture and anticipated benefits that together make this smart dam system a pivotal advancement in water resource management. Aqua Sentinel Stands not merely as a project but as a testament to our commitment to building a sustainable future.

1.1 Current Scenario:

The current scenario in Nepal paints a grim picture of the consequences of natural disasters, particularly floods. Recent events have resulted in significant loss of life, destruction of critical infrastructure, and economic setbacks. The need for a proactive and adaptive solution is evident, and this is where the Aqua Sentinel Smart Dam System steps in.

Some incidents in Nepal that further solidifies the position of Aqua Sentinel Dam System are as follows:

i. Destruction caused by floodwaters in Melamchi bazaar:

Floods triggered by incessant rain since Monday have caused massive damage to life and property in Melamchi Bazaar of Sindhupalchok district.

According to the District Police Office, at least seven people have died while the floods have damaged scores of houses and swept away two concrete motorable bridges and around half a dozen suspension bridges in the district.

Eight trout fish farms, Nakote bridge, farmlands at Timbu, Chanaute Bazar, Amahyalmo Buspark and City Park have also been swept away by the floods.

The floods have also damaged Helambu-based police office, Chinese camp and Armed Police Force camp of the Melamchi Drinking Water Project, said police. (The Kathmandu Post, 2021)

Figure 1: Destruction caused by floodwaters in Melamchi bazaar.

ii. Landslide in Arun III dam highlights the threats to Nepal's Hydropower projects:

A landslide that occurred early Sunday morning blocked the Arun river at Phaksinda in Makalu Rural Municipality, Sankhuwasabha. The landslide took place at the construction site of the Arun III hydroelectric power project.

The landslide debris blocked the diversion tunnel of the project and filled the 26-metre high dam creating a nearly two kilometres long lake, raising the risk of flash floods that could engulf villages downstream. (The Kathmandu Post, 2023)

Figure 2: Water overflowing after the diversion channel is blocked at the Arun III dam site on Sunday.

Nepal is currently grappling with the aftermath of natural disasters, particularly recent floods and landslides that have claimed lives and wreaked havoc on infrastructure. These events underscore the urgent need for innovative solutions to mitigate the impact of such disasters.

In such situations, the application of automation in dams seemed as one of the most reliable sources to us for mitigating the impacts of these disasters. As the world has entered an era of automation, every activity from industrial level to household level have been automated. In such situation, automation in dam system not only helps in supervision for 24/7 preventing the situations of overflow but also makes the dam system free from human errors.

1.2 Problem statement and project as solution:

Flood, which has become more intense and sharper due to climate change, poses a significant threat to Nepal, that has resulted in the loss of lives, destruction of infrastructure, and damage to agricultural lands. The inadequate management of water resources has led to drought conditions, further compounding the challenges faced by the nation.

The incidents like flood water in Melamchi Bazaar, Landslide in Arun Dam III etc. highlights the immense existing problem in the dam infrastructure of Nepal. The unsupervised dam, improper infrastructure, inability of the infrastructure to adapt to the constantly changing climate, absence of warning system etc. has become the highlight of the Nepal's dam structure. Similarly, Nepal is a country with many glacial lakes. In 1996, the Water and Energy Commission Secretariat (WECS) of Nepal reported that five lakes were potentially dangerous, namely, Dig Tsho, Imja, Lower Barun, Tsho Rolpa, and Thulagi, all lying above 4100 m. Such potentially dangerous lakes have high chances of causing glacial eruption and sweeping away the settlement areas near it along with the people. In the scenario of agriculture, outburst of water from reservoirs, droughts etc. has destroyed hundreds and thousands of agricultural lands turning it into barren lands.

In response to this, the Aqua Sentinel Smart Dam System is introduced as a comprehensive solution. Our project not only aims to manage the overflow of water resources in reservoirs but also redirect excess water for the purpose of agricultural irrigation, preserving critical infrastructure like bridges. The Aqua Sentinel Smart Dam System integrates various elements to ensure optimal water storage, controlled release, and precision irrigation for agriculture. With our project we can redirect the water from glacial lakes towards dam for storage, storing the water for a period of time then releasing it whenever there is excess water in the dam. We can also redirect the water for irrigation purposes saving the agricultural plot from cases like drought. The alert system and automated bridge system from our project can save a lot of life and infrastructure from getting destroyed. Similarly, we can monitor the activities of the dam using mobile phones which helps to prevent any kind of unfortunate case. Our project Aqua Sentinel Dam System stands as a commitment to building a sustainable future in the face of unpredictable challenges.

1.3 Aim and Objectives:

The aim of the Aqua Sentinel Dam System is to establish an intelligent and adaptive water dam management infrastructure that addresses the following challenges i.e. water storage, disaster relief system, and agriculture irrigation. This system will serve as a multifaceted solution addressing not only disaster preparedness and response but also sustainable water management and optimized agriculture irrigation.

i. Major Aim:

The major aim of our project is to develop an integrated disaster management system capable of real-time monitoring and early detection of natural disasters like floods, glacier eruption etc.

ii. Objectives:

Other objectives of our project include:

a. Efficient water storage and release:

Design and implement a smart dam system for efficient water storage during non-peak periods and release controlled water during high demand or overflow emergency situation.

b. Agriculture Irrigation:

Deploy sensors and monitoring systems to measure soil moisture levels in agricultural areas. Also develop an irrigation control system that optimally manages water resources, ensuring precision irrigation based on real time agricultural needs.

c. Real time monitoring:

Develop a website for monitoring the alert status, bridge status, dam status and agricultural status of the project using esp32.

d. Automation of bridge:

Deploy sensor to monitor the water release from the dam and lift the bridge above the released water to prevent the bridge from getting swept.

2. Background:

This chapter delves into the fundamental aspects of the Aqua Sentinel Smart Dam System, providing insights into its purpose, functionalities, operational mechanisms, design, and resource requirements.

2.1 System Overview:

The Aqua Sentinel Smart Dam System is an innovative and adaptive water management infrastructure designed to address the complex challenges posed by climate change and the need for sustainable resource management. At its core, the system aims to optimize water storage, provide controlled release during peak demand or emergencies, and facilitate precision irrigation for agriculture.

Our system uses Arduino Uno as the central micro controller that connects all the sensors and actuators. Our system is composed of various sensors and actuators like ultrasonic sensors, soil moisture sensors, LEDs, LCD, servo motor etc. for input and output of data. At the core of our system lies the dam management section, almost all the processes start from this section. This section is responsible for processes like storage of water, checking the water level, controlled release of water, providing input for other sections like alarm section, bridge automation section and agriculture section etc.

The system operates as a multifaceted solution to several challenges faced by Nepal. The key functionalities include:

i. Infrastructure Protection:

The system incorporates an automated bridge management system to safeguard critical infrastructure during floods.

ii. Efficient Water Storage and Controlled Release:

The system intelligently manages the storage of water during non-peak periods, ensuring optimal utilization of resources. In high-demand situations or emergency overflow conditions, the system releases water in a controlled manner, preventing catastrophic flooding.

iii. Agriculture Irrigation:

Utilizing sensors and monitoring systems, the system measures soil moisture levels in agricultural areas and deploys an irrigation control system for precision watering based on real-time agricultural needs.

iv. Real time monitoring:

The real time monitoring of the status of dam, bridge, alarm and agricultural plot land is one of the major functionalities of our project.

As for the working our system, our system pumps water into the dam using the water pump. Then using ultrasonic sensor we sense the water level in the dam. Once the water level reaches its maximum limit, the alarm system is triggered as well with the bridge being lifted up. Then the dam valve is lifted up using servo motor. After this process, the soil moisture sensor checks the water storage. If there is absence of water in the storage, then the water pump connected with relay module draws the water into the storage. For the irrigation section, the soil moisture sensor checks the soil and pumps the water from the storage in case of dry soil.

2.2 Design Diagrams:

Our project model consists of various factors like mountains, dam, river, settlement area, agricultural land and alarm tower. The mountains are made up of paper stacked upon each other. From the bottom of the mountains, we have inserted a pipe into the dam through which water is pumped out of using water pump. The dam is made out of plastic container and sealed with water proof sticker to prevent the leakage of water. The dam consists of a valve through which water is released out of. The valve is lifted up using servo motor. Similarly, the base for our model is made out of cardboard boxes stacked upon each other. The river area is sealed with water proof sticker as well. Directly above the river lies the bridge which is lifted up using servo motor. The settlement areas and alarm tower are made out of cardboard boxes. The alarm tower consists of LED and buzzer for alerting purpose whereas the agricultural plot consists of two plots. One plot consists of dry soil whereas another one consists of moist soil. The storage facility for the agricultural purpose is made out of plastic bottle.

i. Model Diagram:

The model diagram for our project is given below:

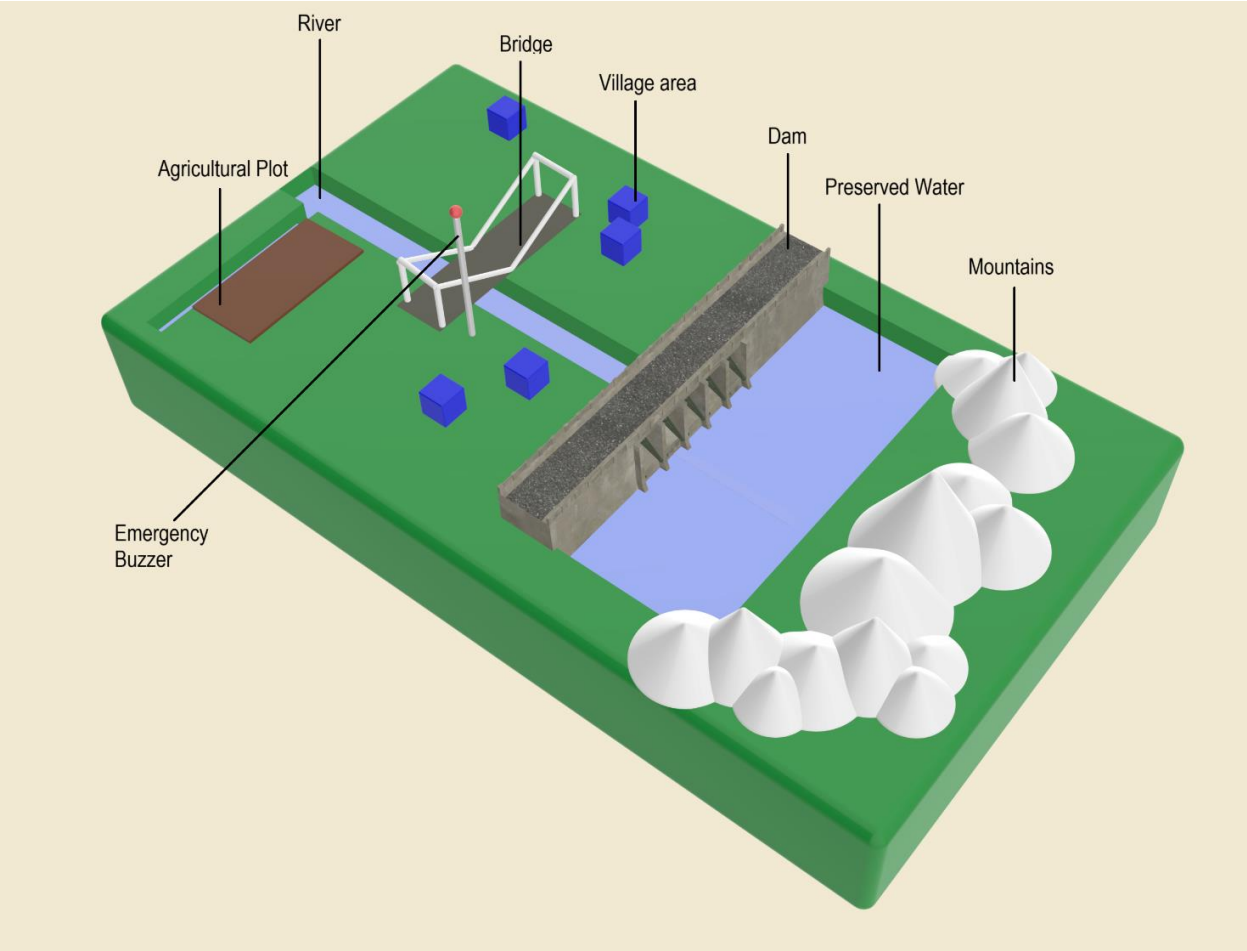

Figure 3: Model Diagram of our project

ii. Circuit Diagram:

The circuit diagram for our project Aqua Sentinel Dam System is as follow:

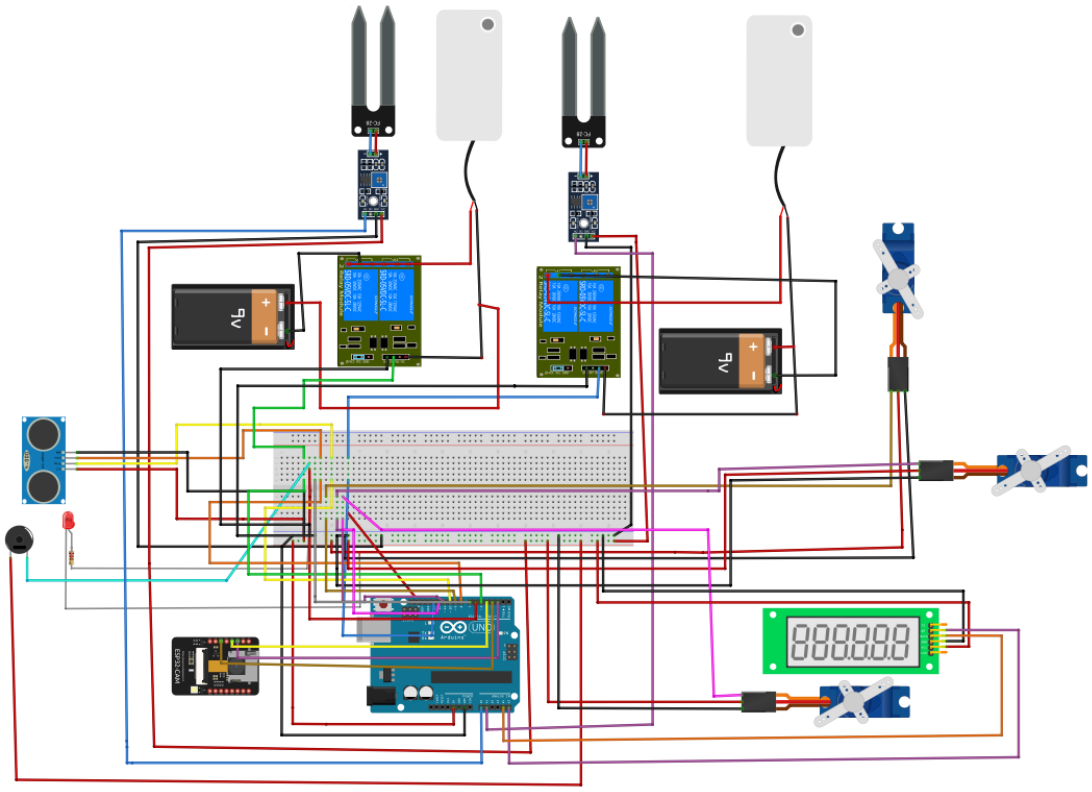

Figure 4: Circuit Diagram of our project

iii. Flow chart:

The flow chart for our project Aqua Sentinel Dam System is as follows:

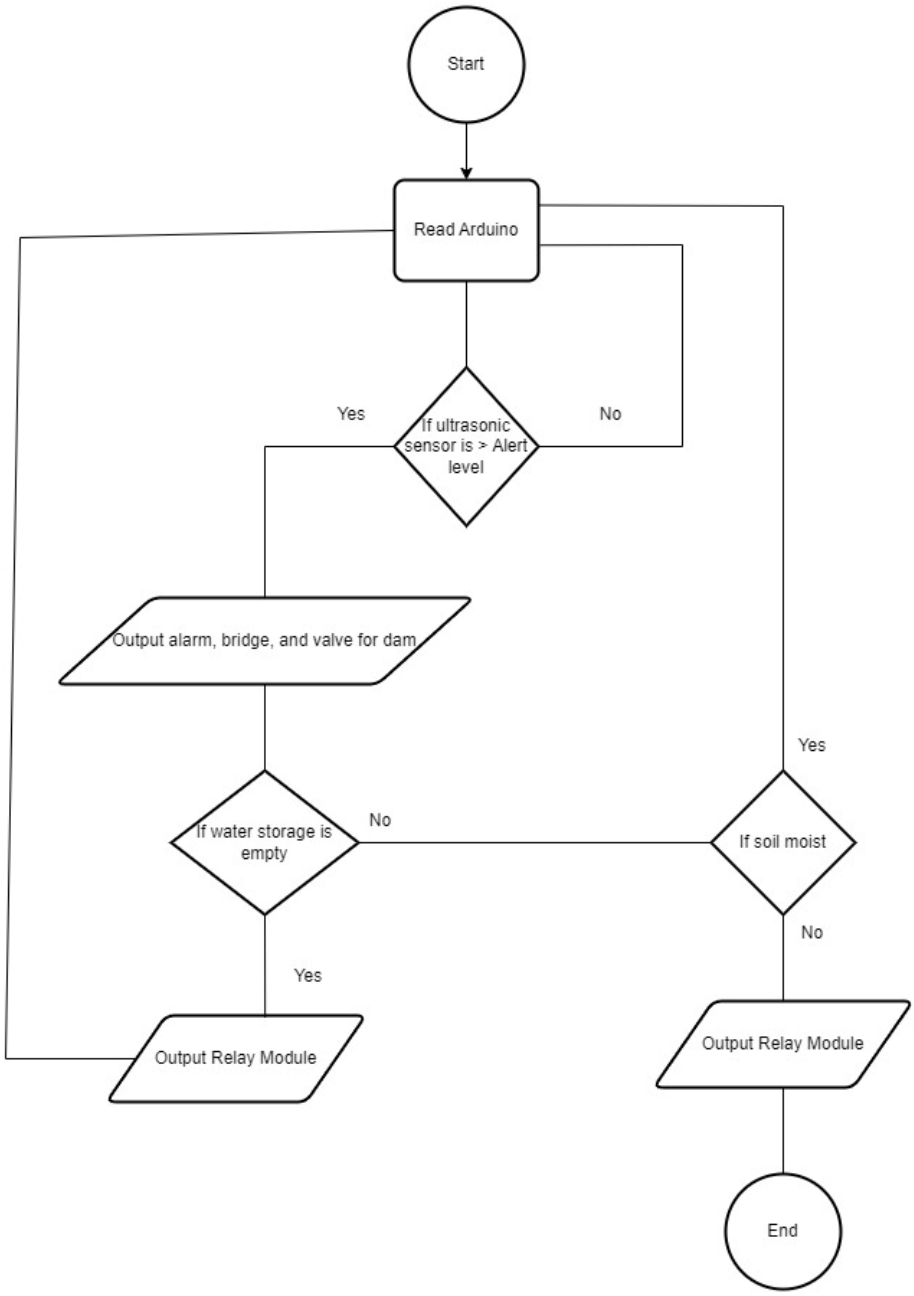

Figure 5: Flowchart of our project

iv. System Architecture:

The system architecture for our project Aqua Sentinel Dam System is as follows:

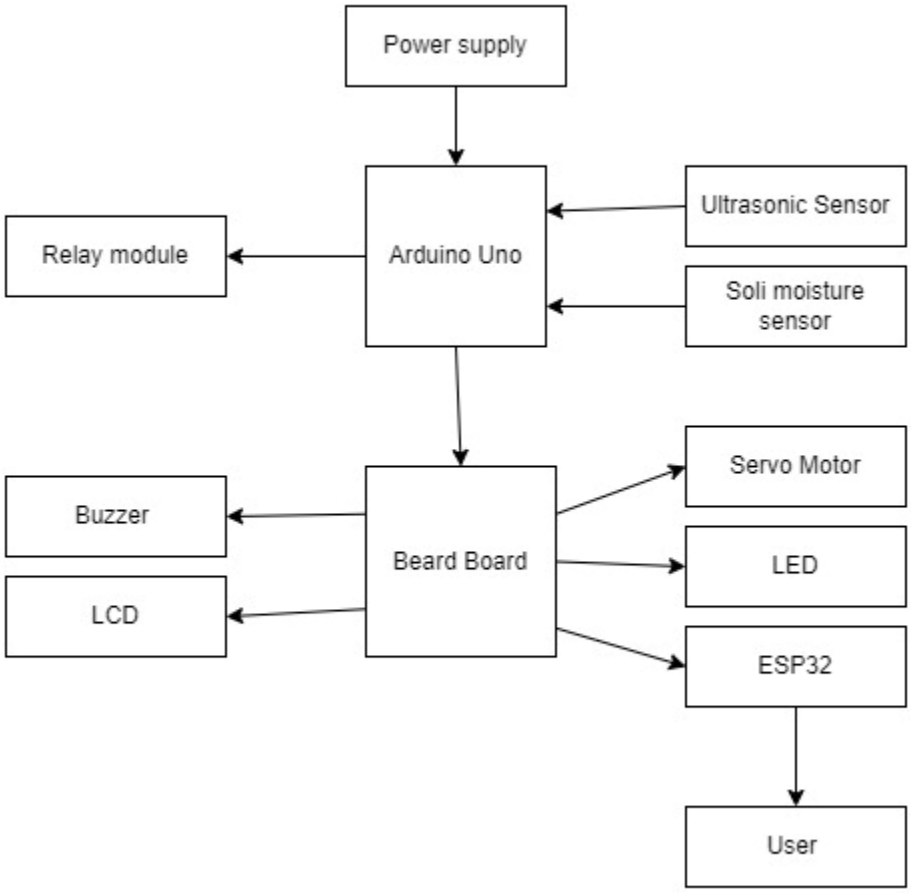

Figure 6: System Architecture of our project

2.3 Requirement Analysis:

The selected hardware and software that are required for the completion of our project are as follows:

i. Hardware resources:

a. Arduino Uno:

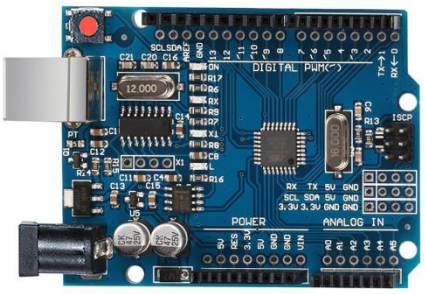

Figure 7: Arduino Uno

- Role: The Arduino Uno is the central microcontroller that processes data from sensors and controls the various components in the system.
- Reason for suitability: Arduino Uno is easy to use, highly cost effective and energy efficient. The platform is also open source and has an extensive user support community.

b. Jumper wire:

Figure 8: Jumper Wire

- Role: Jumper wires are used to establish connections between different components, such as sensors, servomotors, and other devices connected to the Arduino.
- Reason for suitability: Little technical knowledge is required to use jumper wire and it is cost effective as well.

c. Ultrasonic sensor:

Figure 9: Ultrasonic Sensor

- Role: Measures the distance of an object (such as water level) by emitting ultrasonic waves and calculating the time taken for the waves to bounce back.
- Reason for suitability: Ultrasonic sensors are highly accurate and can detect very small alteration in position.

d. Soil Moisture Sensor:

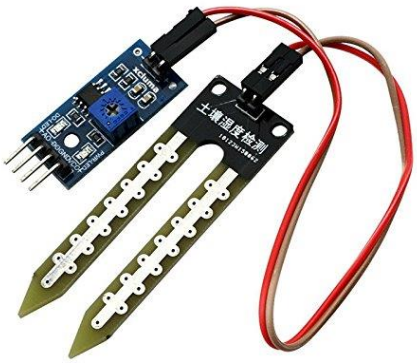

Figure 10: Soil Moisture Sensor

- Role: It is used to sense the water level in soil.
- Reason for suitability: Soil moisture sensor aids in good irrigation management.

e. Buzzer:

Figure 11: Buzzer

- Role: Alerts the surrounding environment when the water level is about to reach the maximum level, indicating a potential flood or overflow situation.
- Reason for suitability: Buzzer uses minimal current and is very cost effective.

f. LED:

Figure 12: LED

- Role: Blinks or lights up when the water level is about to reach the maximum level, serving as a visual indicator.
- Reason for suitability: LED uses minimal current and is very cost effective.

g. Servo Motor:

Figure 13: Servo Motor

- Role:
    - Lifts the bridge when the water is sensed, allowing for water passage.
    - Lifts the gate of the dam when the water level rises to the maximum, controlling the release of water.
- Reason for suitability: Servo motor is efficient and accurate.

h. Bread Board:

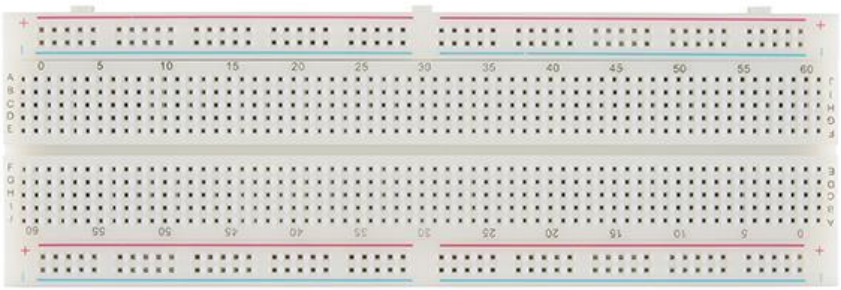

Figure 14: Bread Board

- Role: Provides a platform for prototyping and connecting various electronic components without soldering.
- Reason for suitability: For getting sufficient pins for connection and connecting Arduino Uno with ESP32-CAM.

i. Water pump motor:

Figure 15: Water Pump Motor.

- Role: Actuates to control the flow or movement of water as needed in the system.
- Reason for suitability: Can pump large volumes of water as per need.

j. USB Cable:

Figure 16: USB Cable

- Role: Connects the Arduino Uno to a computer or power source for programming and powering the system.
- Reason for suitability: Easy to connect and disconnect and is compatible with wide range of devices.

k. LCD:

Figure 17: LCD

- Role: Displays the water level information, making it visible to users or operators.
- Reason for suitability: Wiring is straightforward, requiring only two data pins to control the LCD.

l. ESP32-CAM:

Figure 18: ESP32-CAM

- Role: Integrating Wi-Fi into the project.
- Reason for suitability: Powerful, versatile and used for internet connection.

m. Relay Module:

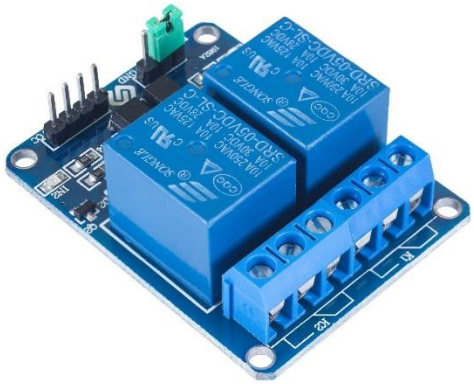

Figure 19: Relay Module

- Role: Switches the connected devices on or off.
- Reason for suitability: Relay module is reliable, durable and of low cost.

n. Resistor:

Figure 20: Resistor

- Role: Used to reduce current flow, adjust signal levels and to divide voltages
- Reason for suitability: Cheap and suitable for controlling the ampere flow from Arduino to LED.

ii. Software Resources:

a. Microsoft word:

Figure 21: Microsoft Word

- Role: This program is utilized for documentation.
- Reason for suitability: User-friendly and provides formatting options.

b. Vectary:

Figure 22: Vectary

- Role: Creating a 3D model of our project.
- Reason for suitability: For visualizing the model to be made for the project.

c. Arduino IDE:

Figure 23: Arduino IDE

- Role: For building code for Arduino.
- Reason for suitability: It is open source software, supports multiple libraries and supports cross platform.

d. Fritzing:

Figure 24: Fritzing

- Role: For creating our project's circuit diagram.
- Reason for suitability: It provides an intuitive and user-friendly interface that allows users to create 3D models.

3. Development:

The step by step process for the development of the project are as follows:

Step 1: Research

- On week 6 of our first semester, we were provided with our coursework to develop an IOT system.
- To choose our IOT system, we had to make an extensive amount of research about which IOT system was feasible and most suitable for us to develop.
- Through an extensive amount of research, we decided on our Aqua Sentinel Dam System.

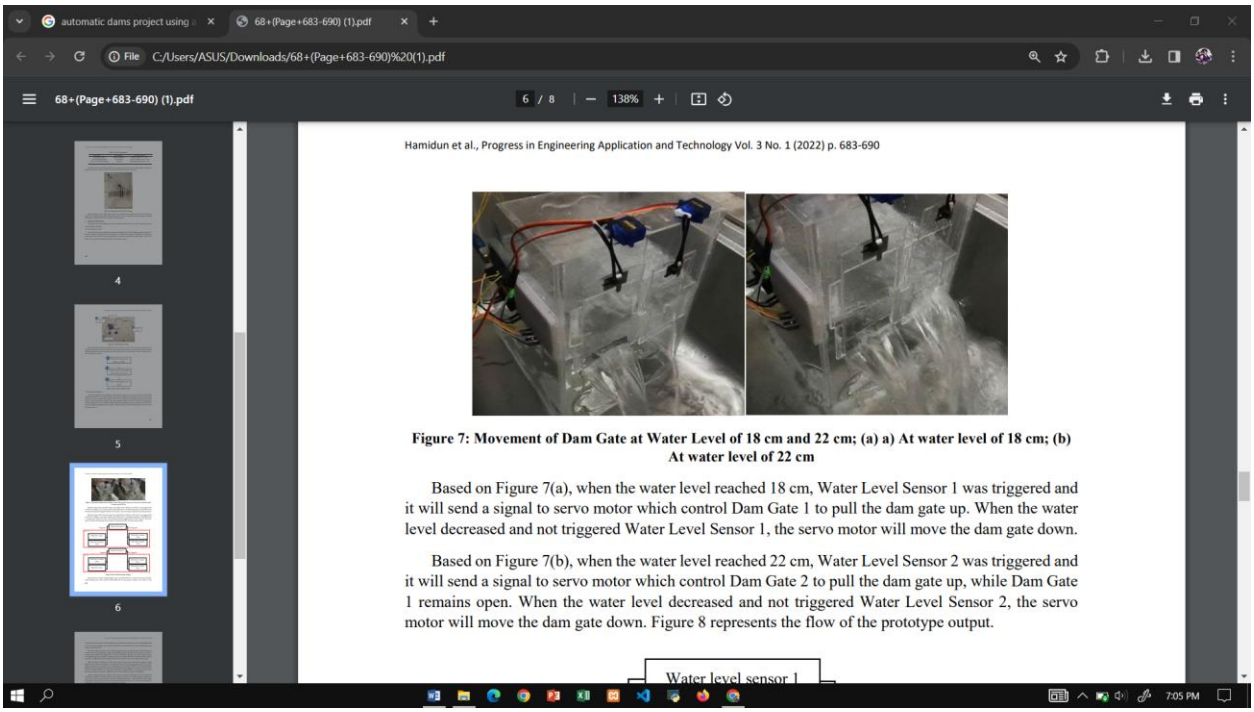

Figure 25: Research of IOT system.

-

Step 2: Requirement analysis

- In order to make our IOT system, we had to inquire about the devices needed to be develop our system.
- We consulted with our module leader for the requirements of our IOT system.

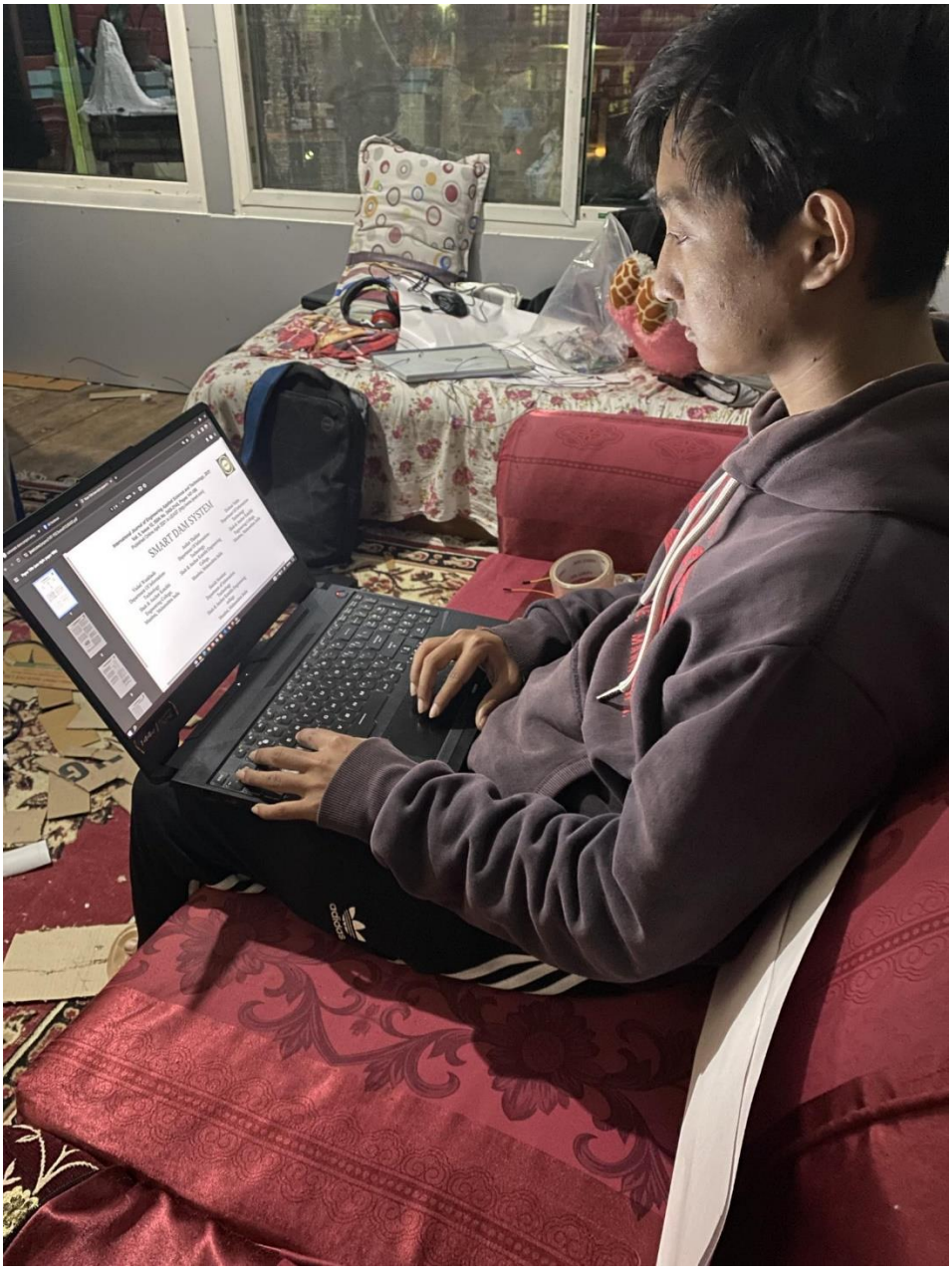

Figure 26: Requirement analysis by group member.

Step 3: Resource collection

- Our college provided us with materials needed for the development or our system.
- We were provided with devices like Arduino Uno, jumper wires, LED, buzzer, water pump motor, sensors, servo motor, bread board etc.
- As for the cardboard boxes, plastic container and other materials, we bought them form various shops.

Figure 27: Collection of resources.

Step 4: Testing of Components.

- After collecting the components from the IT department, we had to test each component to make sure that all the components were working.
- The testing of components was done by our group member Prabal Gurung, Prabin Thapa and Rahul G.C.

Figure 28: Testing of single component at a time.

Step 5: Design of the system

- Before developing our Aqua Sentinel Dam System, we had to make system diagram.
- Circuit diagram was created using Tinker Cad, flow chart was made using Draw.io and model diagram was using vectary.

Figure 29: Reference for design of system.

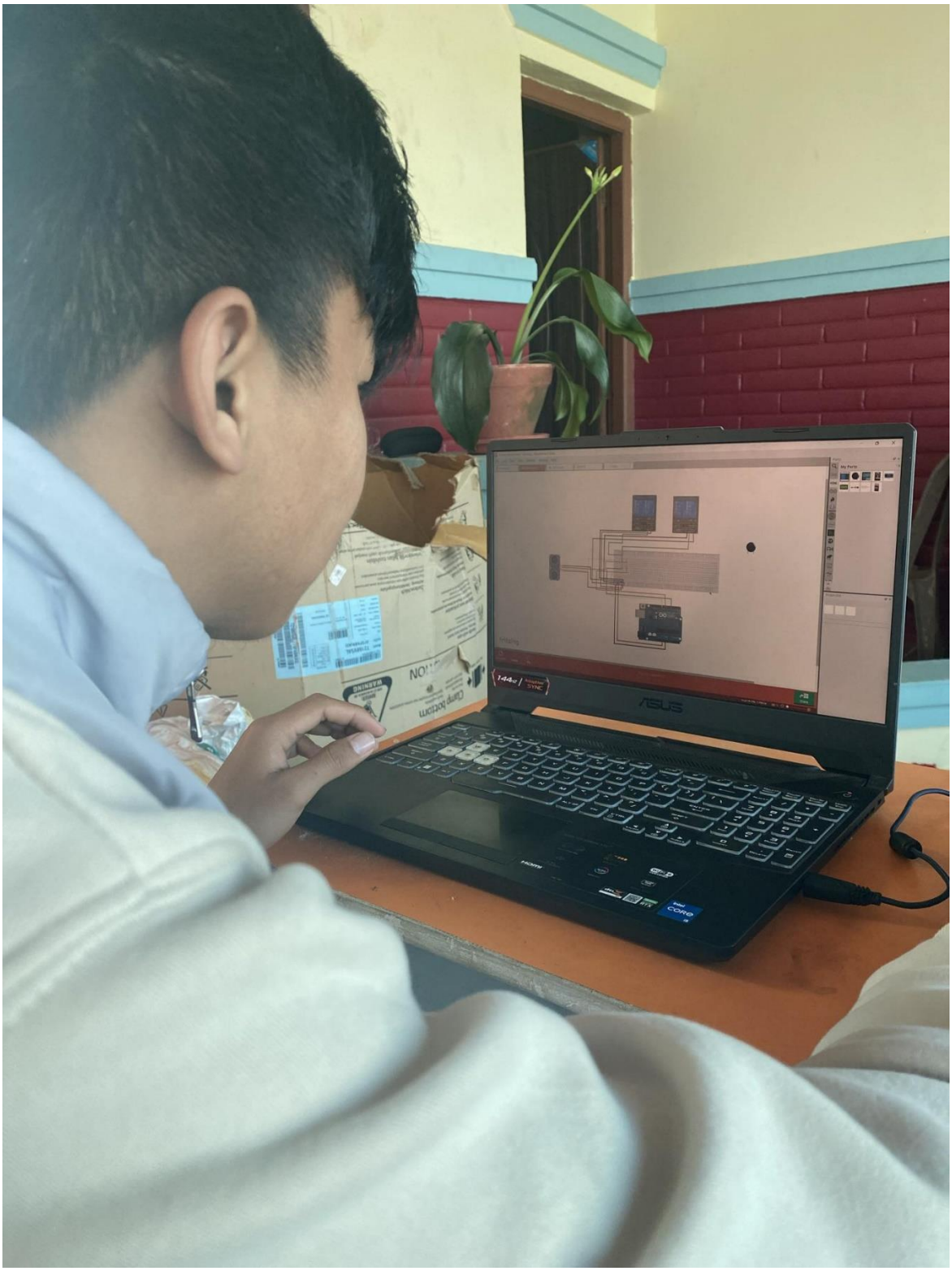

Figure 30: Designing circuit diagram in fritzing.

Step 6: Development of system

- After the design of our system was made, we started the development of our system.
- Our system was divided into 4 sections namely, dam section, bridge section, environment model and agriculture section. Each section was provided to four of the members for development.
- The coding for our system was done in a parallel manner to the development of our system.
- During this step, we encountered problems in the development of our environment as well as coding section.

Figure 31: Development of system.

Figure 32 : Development of system II

Step 7: Review and update

- The developed sections were reviewed by all the members of the group.
- Each section was provided with ways to improve it by the members.
- The reviewed sections were updated according to the suggestions.
- During the review of the code of system, we encountered a problem in connecting esp32 cam with Arduino Uno. This was later tackled with the help of our module leader.

Figure 33: Review and update of system

Figure 34 : Review and update of system II

Step 8: Integration of system

- After all the sections were developed, we integrated the sections together.
- The integration of the devices into the environment was also done in this part.
- During the integration, we encountered a problem with jumper wire. The length of the jumper wires was not enough to connect all our devices across the system.
- This problem was mitigated as we did solder for connecting multiple jumper wires.

Figure 35: Integration of system.

Step 9: Testing of whole system

- The testing of the system was done after all the integration was completed.
- During the testing, our major concern was in the dam section as there were chances of occurrence of leakage which could possibly destroy the whole ecosystem.

Figure 36: Testing of whole system.

4. Results and findings:

4.1 Project Outcome:

At the end of our project, the expected outcome is a disaster relief system that prevents catastrophes like flood, glacier eruption etc. preventing the loss of lives and properties. Our final product Aqua Sentinel Smart Dam System will utilize Arduino Uno for controlling all the other components, Jumper wires for connection, sensors for measuring water levels, buzzers and LCD for warning purposes, water pump for pumping water into the dam and water storage for agricultural purpose and servo motor for lifting bridge and valve for dam. The final product will be able to sense the water level in the dam and prevent overflow by opening the valve. In case of overflow, the alarm will be triggered along with the buzzer and the bridge will be lifted up. Similarly, the excessive water will be pumped towards the agricultural plot land.

The picture for our completed project model is given below:

Figure 37: Completed Project.

4.2 Test cases:

The testing of our project is given below:

i. Test 1:

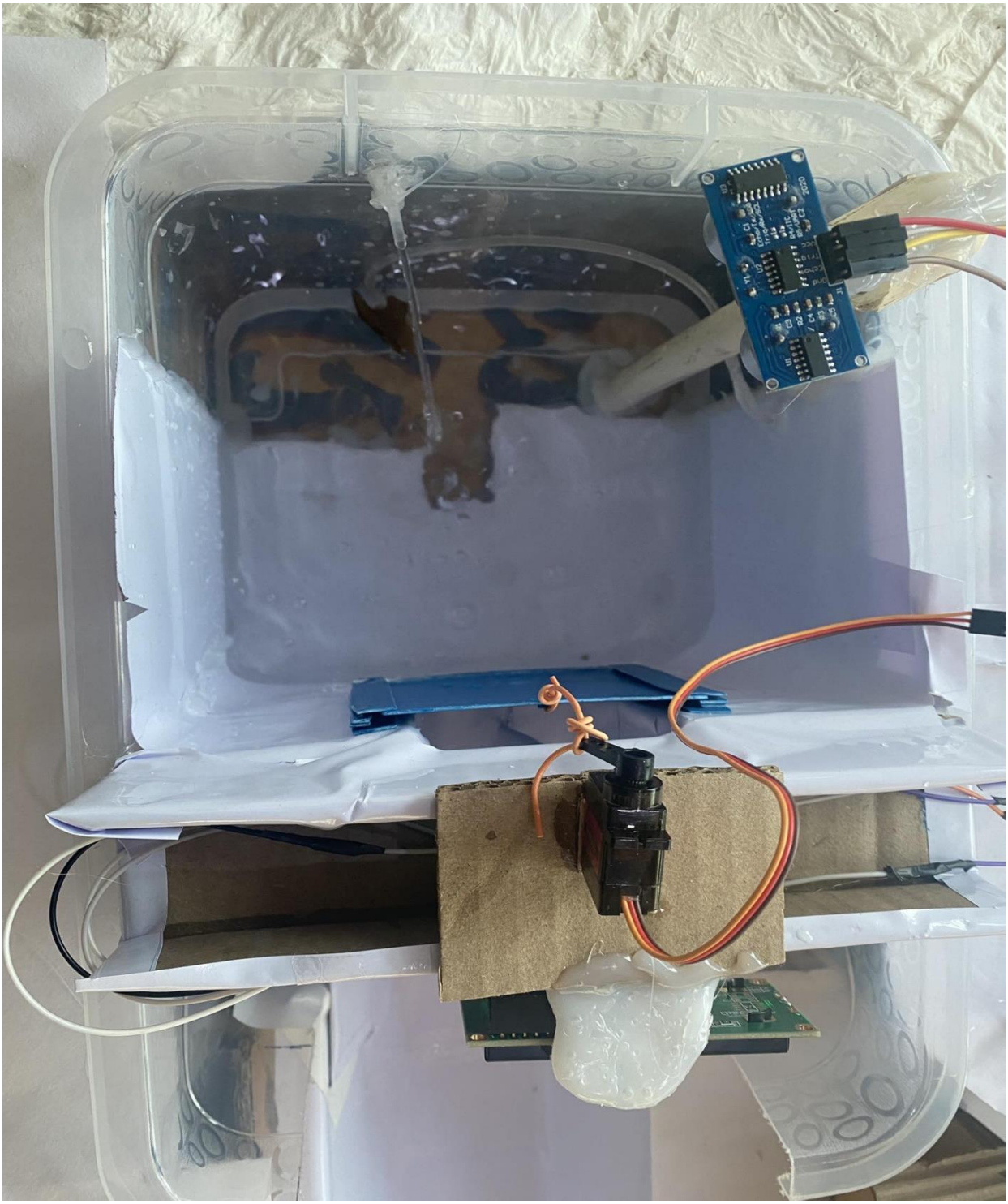

Figure 38: Testing for water pump.

Table 1: Test case no.1

Objective	To check whether the water pumps into the dam or not.
Action	Connect the battery with the water pump and provide a water source to the water pump.
Expected Result	The water should be pumped through the pipeline.
Actual Outcome	The water is pumped into the dam through pipe line.
Conclusion	Test successful.

ii. Test 2:

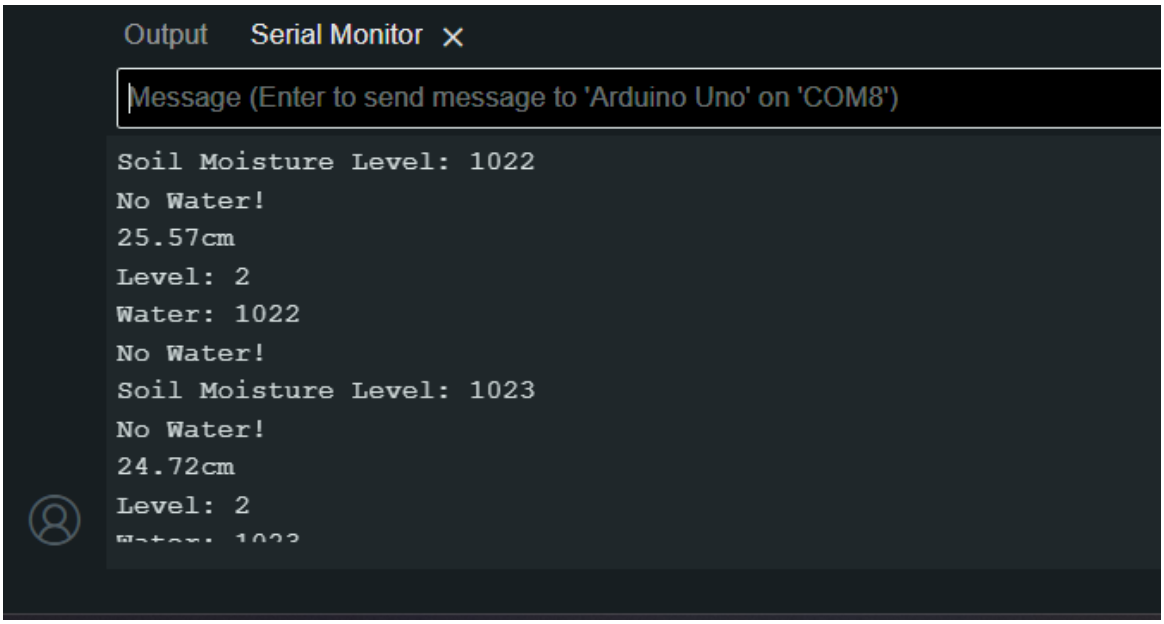

Figure 39: Test for ultrasonic sensor

Table 2: Test case no.2

Objective	To check whether the ultrasonic sensor senses the water level or not.
Action	Connect the Arduino Uno with the laptop and upload and run the code. Constantly check the serial monitor for the changing water level.
Expected Result	The ultrasonic sensor should sense the changing water level.
Actual Outcome	The ultrasonic sensor senses the changing water level in the dam.
Conclusion	Test successful.

iii. Test 3:

Figure 40: Test for valve of dam.

Table 3:Test case no.3

Objective	To check whether the valve is opened or not during excess of water
Action	Connect the Arduino Uno with the laptop and upload and run the code.
Expected Result	The valve should be lifted up by the servo motor.
Actual Outcome	The valve is lifted up by the servo motor and the excess water is released.
Conclusion	Test successful.

iv. Test 4:

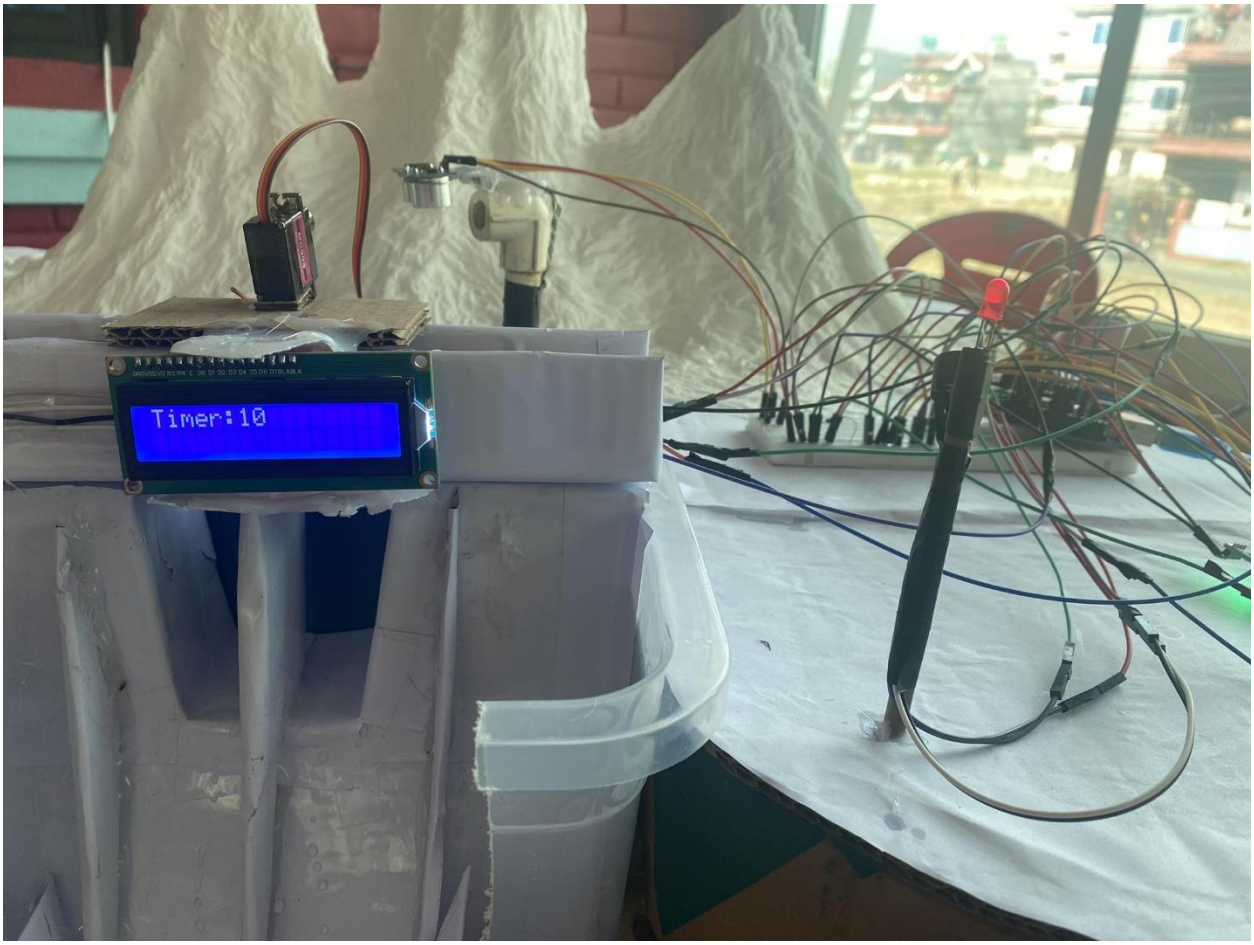

Figure 41: Test for alarm system.

Table 4: Test case no.4

Objective	To check whether the alarm system works properly or not.
Action	Connect the Arduino Uno with the laptop and upload and run the code.
Expected Result	The LED and buzzer should be triggered on the opening of the valve.
Actual Outcome	The LED and buzzer are triggered during the release of water from the dam.
Conclusion	Test successful.

v. Test 5:

Figure 42: Test for bridge.

Table 5: Test case no.5

Objective	To check whether the bridge is lifted up or not.
Action	Connect the Arduino Uno with the laptop and upload and run the code.
Expected Result	The bridge should be lifted up by the servo motor after the release of excess water.
Actual Outcome	The bridge is lifted up by the servo motor after the release of water.
Conclusion	Test successful.

vi. Test 6:

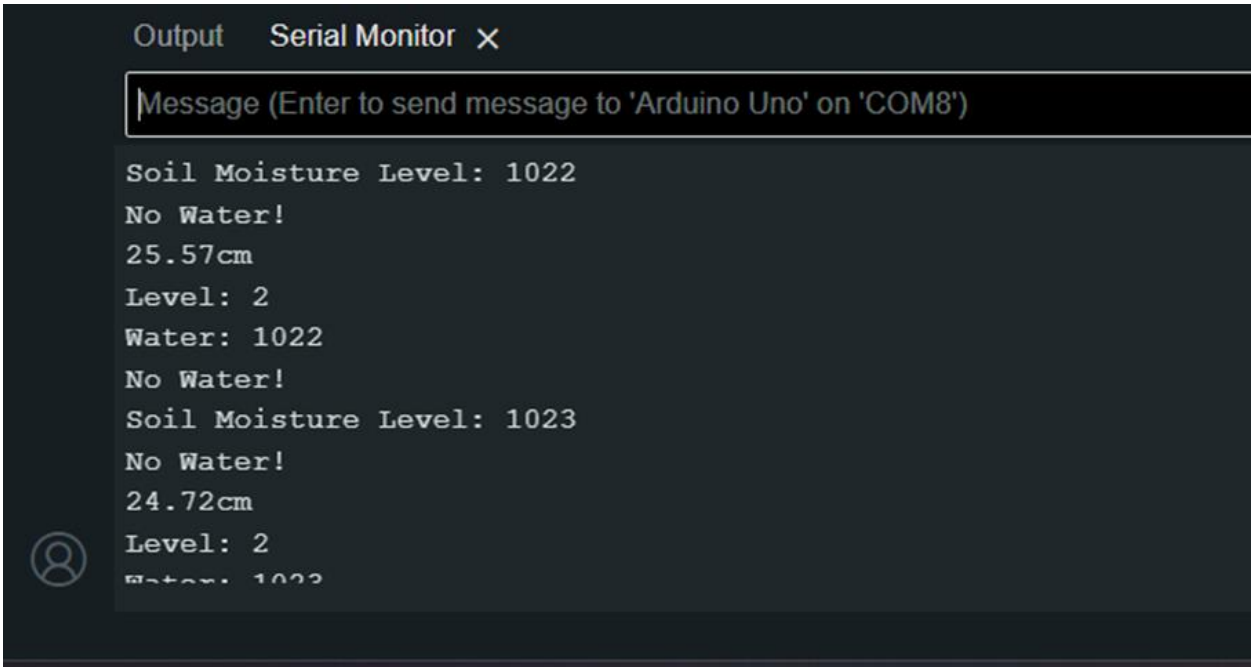

Figure 43: Test for soil moisture.

Table 6: Test case no.6

Objective	To check the soil moisture level of the agricultural plot land.
Action	Connect the Arduino Uno with the laptop and upload and run the code.
Expected Result	The soil moisture sensor should sense the moisture level of the soil.
Actual Outcome	The soil moisture sensor detects the moisture level of soil.
Conclusion	Test successful.

vii. Test 7:

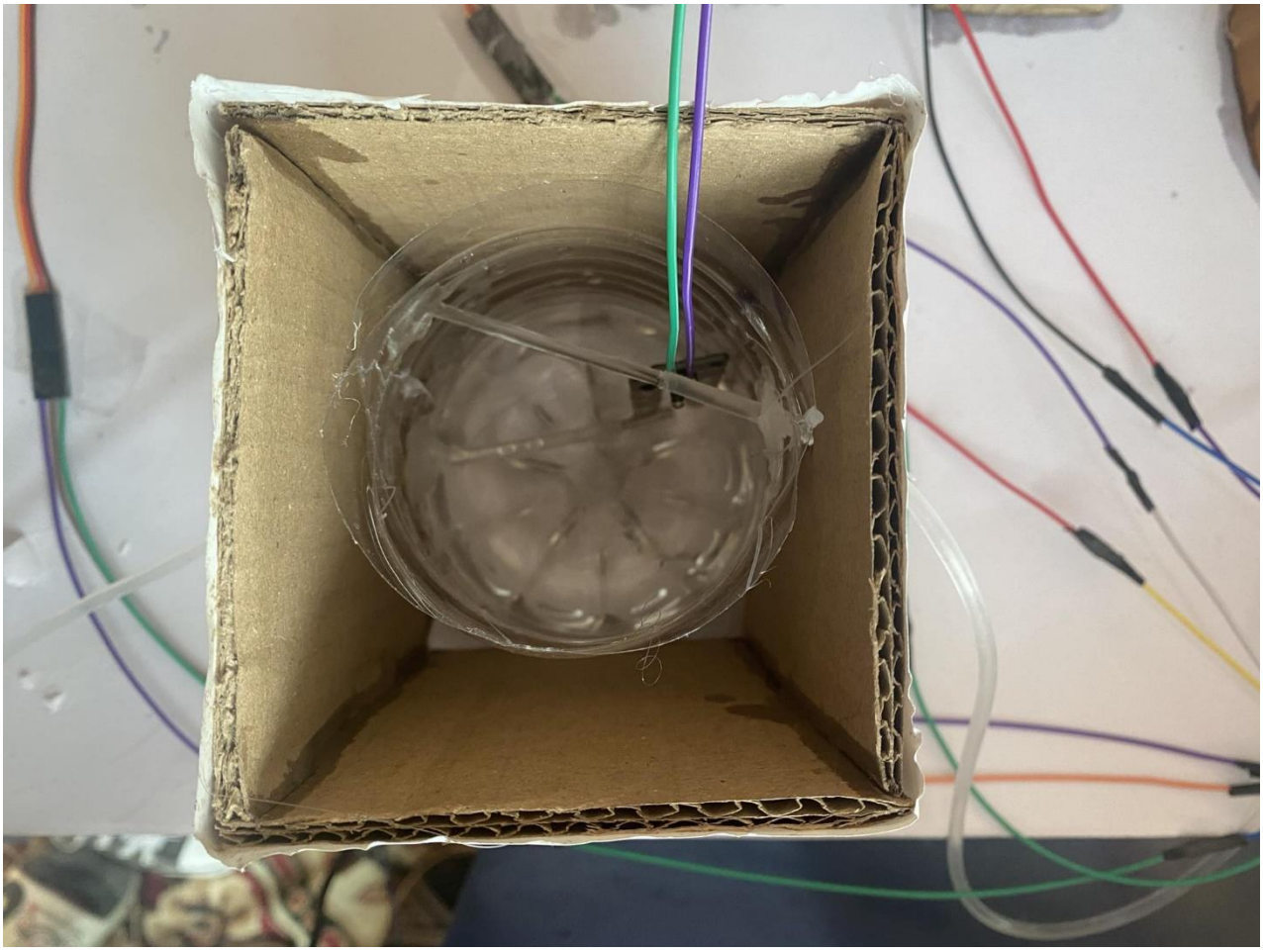

Figure 44: Test for water pump to agricultural plot.

Table 7: Test case no.7

Objective	To pump the water into the water container for agricultural land.
Action	Connect the Arduino Uno with the laptop and upload and run the code.
Expected Result	The water pump should dispense the water into the water container designated for irrigation.
Actual Outcome	The water pump pumps the water into the container.
Conclusion	Test successful.

viii. Test 8:

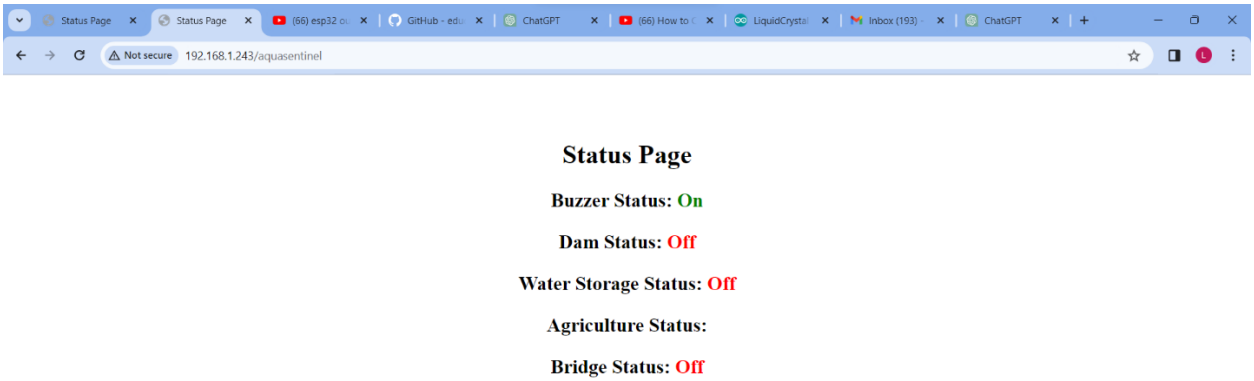

Figure 45: Test for guest perspective of website.

Table 8: Test case no.8

Objective	To test the real live monitoring of our project from guest perspective.
Action	Enter the IP address for our website from any device and log in to guest user.
Expected Result	The user should be able to monitor the status of project from their devices through website.
Actual Outcome	The user is able to see the status of the various aspects of project through the website.
Conclusion	Test successful.

5. Future works:

Our project Aqua Sentinel Dam System is a project that integrates the environmental and technological aspects into one combined form. It encompasses the aspect of sustainability and reusability. However, our project is not perfect in itself. Aqua Sentinel Dam System has a lot of room for improvements. Some future works that can be added in the project in order to make it better, sustainable and usable are as follows:

I. Generation of Hydroelectric power: One aspect that can be added in our project in the future is hydropower. A turbine can be added towards the section of dam in order to generate hydropower through the release of the water. This hydropower can become a source of energy for the settlement areas as well as for other aspects making the project more sustainable.

II. Real time monitoring: Currently our project only has real time monitoring of our project. However, our project does not store the real time monitoring data in any form of cloud. In the future, cloud storage services like DropBox, OneDrive etc. can be used to keep record of the data of real time monitoring. Also, we can scale our project by adding functionalities for admin to directly control the project through web app for maintenance and checking.

III. Automated barriers for landslides: Another aspect that can be added in our project is automated barriers for landslides. We can use geotechnical sensors to detect the soil movement and make automated barriers to barricade the debris from falling towards the settlement areas.

IV. Weather forecasting: Weather forecasting is another functionality that can be added in our project. As rainfall is also one of the major reasons for outburst of water, weather forecasting will prove to be a valuable resource to manage the dam.

6. Conclusion:

In conclusion, our project Aqua Sentinel Smart Dam System aims to create a resilient and adaptive water management system that addresses the current challenges of water storage, controlled release, and agricultural irrigation. As the issues concerning environmental problems like floods are on the rise in Nepal, Aqua Sentinel Smart Dam System emerges as a boon.

Our project Aqua Sentinel Smart Dam System is able to tackle on the problem of unmanaged dam system that has been burdening and causing loss of lives and properties. By incorporating Arduino Device system will not only ensures efficient water usage but also contribute to environmental sustainability and community safety. Our projects strength lies in its positive effect towards the environment. As our project seamlessly incorporates technology in an eco-friendly manner, it has zero to no negative effect in the eco system. On the other hand, it preserves the eco system from deterioration.

This coursework was very fun and was a challenging one. Doing this coursework was like a roller-coaster there were a lot of ups and downs. However, with a lot of effort and team work we were able to accomplish our work. We encountered problems during the development of environment for our project, during the coding phase as well as during the integration phase of our system. But we tackled our way through these problems making these problems a stepping stone to success. Time management was a crucial part of our project. Without proper time management, we would not have been able to complete our work in time.

7. References:

The Kathmandu Post, 2021. *Destruction caused by floodwaters in Melamchi Bazaar.* [Online]
Available at: https://kathmandupost.com/visual-stories/2021/06/16/destruction-caused-by-floodwaters-in-melamchi-bazaar
[Accessed 7 January 2024].

The Kathmandu Post, 2023. *Lanslide in Arun III dam highlights threats to Nepal's hydropower projects.* [Online]
Available at: https://kathmandupost.com/climate-environment/2023/05/01/landslide-in-arun-iii-dam-highlights-threats-to-nepal-s-hydropower-projects
[Accessed 7 January 2024].

8. Appendix:

8.1 Source Code:

Source Code for AquaSentinel.ino:

```
#include <Wire.h>

#include <Servo.h>

#include <LiquidCrystal_I2C.h>


// Pin assignments

int trigPin = 10;       // Trigger pin for ultrasonic sensor

int echoPin = 8;        // Echo pin for ultrasonic sensor

const int hook = 2;     // Hook control pin


// LCD configuration

LiquidCrystal_I2C lcd(0x27, 16, 2);


// Alarm pin

int alarm = 6;


// Relay control pins

int relayStoragePin = 13;

int relayAgriculturePin = 4;


// Analog sensor pins

int waterStorageSensor = A0;

int agricultureSensor = A1;
```

```
Servo bridgeServoRight;
Servo bridgeServoLeft;
Servo dampServo;
Servo agricultureServo;

// Ultrasonic sensor variables
float duration_us, distance_cm;

// Flag to ensure alarm only triggers once
int alarmOnce = 2;

void setup() {
  Serial.begin(9600);

  // Attach servo motors
  dampServo.attach(9);
  agricultureServo.attach(12);
  bridgeServoRight.attach(3);
  bridgeServoLeft.attach(11);

  // Initialize LCD
  lcd.init();
  lcd.backlight();
```

```
  // Pin mode configurations
  pinMode(echoPin, INPUT);
  pinMode(waterStorageSensor, INPUT);
  pinMode(agricultureSensor, INPUT);
  pinMode(hook, OUTPUT);

  // Alarm setup
  pinMode(alarm, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(relayAgriculturePin, OUTPUT);
  pinMode(relayStoragePin, OUTPUT);

  // Initial states
  digitalWrite(alarm, HIGH);
  digitalWrite(trigPin, HIGH);
  digitalWrite(hook, HIGH);

 // Initial LCD message
  lcd.setCursor(0, 0);
  lcd.print("Dam: Close" );

  // Initial servo positions
  dampServo.write(0);
  bridgeServoLeft.write(20);
```

```
  bridgeServoLeft.write(0);
}

void loop() {
  // Ultrasonic sensor reading
  digitalWrite(alarm, HIGH);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  //calculation
  duration_us = pulseIn(echoPin, HIGH);
  distance_cm = 0.017 * duration_us;

  Serial.print(distance_cm);
  Serial.println("cm");

  // Control hook based on distance
  if (distance_cm <= 12){
    digitalWrite(hook, HIGH);
    delay(4000);
  } else {
    digitalWrite(hook, LOW);
    delay(4000);
```

```
  }

  // Call additional features based on distance and sensor readings
  features(distance_cm, waterStorageSensor);
}

// Function to handle features
void features(int distance_cm, int waterStorageSensor){
  if (distance_cm <= 28){
    damLevel(distance_cm);
  } else if (distance_cm == 0 or distance_cm > 20) {
    Serial.println("ALERT OVERFLOW");
  }

  agriculturePath();

  if (waterStorageSensor < 400) {
    agriculture();
  }
}

// Function to control dam level based on distance
void damLevel(int distance_cm) {
  int currentLevel;
```

```
 int alertLevel = 8;


 // Calculate dam water level
 currentLevel = 14 - (distance_cm / 2);
 Serial.print("Level: ");
 Serial.println(currentLevel);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Dam: Closed ");


 // Initialize alarm flag
 if (alarmOnce == 2){
   alarmOnce = 1;
 }


 // Check if dam level is above alert level
 if (currentLevel >= alertLevel){
   // Trigger alarm and open the dam
   if (alarmOnce == 1){
       for (int z = 0; z <= 10; z++) {
       lcd.setCursor(0, 0);
       lcd.print("Timer:" );
       lcd.print(z);
       lcd.print("      ");
```

```
      delay(800);
    }
    for (int i = 1; i <= 12; i++) {
      digitalWrite(alarm, LOW);
      delay(150);
      digitalWrite(alarm, HIGH);
      delay(150);
    }
    for (int angle = 0; angle <= 40; angle++) {
      bridgeServoLeft.write(angle);
      bridgeServoRight.write(60 - angle);
      delay(50);
    }

    delay(2000);
    alarmOnce = 0;
  }
  dampServo.write(180);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Dam: Open");

} else if (currentLevel < alertLevel){
  // Close the dam and reset the alarm
```

```
    dampServo.write(0);

    for (int angle = 40; angle >= 0; angle--) {
      bridgeServoLeft.write(angle);
      bridgeServoRight.write(60 - angle);
      delay(50);
    }

    alarmOnce = 1;
  }
}

// Function to control water storage based on sensor readings
void agriculturePath() {
  int dryThreshold = 500;
  int water = analogRead(waterStorageSensor);

  Serial.print("Water: ");
  Serial.println(water);

  // Check if water level is below the threshold
  if (water > dryThreshold) {
    Serial.println("No Water!");
    digitalWrite(relayStoragePin, HIGH);
```

```
  } else {
    digitalWrite(relayStoragePin, LOW);
    Serial.println("LIMITED");
  }

  delay(1000);
}

// Function to control agriculture based on soil moisture
void agriculture() {
  int soilValue = analogRead(agricultureSensor);

  Serial.print("Soil Moisture Level: ");
  Serial.println(soilValue);

  // Check if soil moisture is below a certain threshold
  if (soilValue > 500) {
    Serial.println("No Water!");
    digitalWrite(relayAgriculturePin, HIGH);
  } else {
    digitalWrite(relayAgriculturePin, LOW);
    Serial.println("LIMITED");
  }
  delay(1000);
```

```
}
```

Source Code for guestConnection.ino

```
#include <WiFi.h>
#include <WebServer.h>

// WiFi credentials
const char* ssid = "Kantipur Party Palace";
const char* password = "9856024196";

// Pin assignments for various components
const int buzzerPin = 12;
const int damPin = 13;
const int waterStoragePin = 14;

// Web server instance
WebServer server(80);

// HTML content for the login page
const char* htmlContent1 = R"=====(
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aqua Sentinel Login</title>
```

```
<style>
  body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(to right, #3498db, #2c3e50);
    margin: 0;
    padding: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
  }

  .login-container {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
    padding: 40px;
    text-align: center;
  }

  h2 {
    color: #3498db;
  }
```

```
    .login-button {
      background-color: #3498db;
      color: #fff;
      border: none;
      padding: 12px 24px;
      margin: 10px;
      border-radius: 5px;
      cursor: pointer;
      font-size: 18px;
      transition: background-color 0.3s;
    }

    .login-button:hover {
      background-color: #2980b9;
    }

    .dam-icon {
      font-size: 50px;
      color: #3498db;
      margin-bottom: 20px;
    }

    /* Responsive styling for smaller screens */
    @media (max-width: 600px) {
```

```
      .login-container {
        width: 80%;
      }
    }
  </style>
</head>
<body>

<div class="login-container">
  <span class="dam-icon">🌊</span>
  <h2>Smart Dam System</h2>

  <button class="login-button" onclick="loginAsAdmin()">Admin</button>
  <button class="login-button" onclick="loginAsGuest()">Guest</button>
</div>

<script>
  function loginAsAdmin() {
    var password = prompt("Enter the password for admin:");
    if (password === "POKEMON") {
      alert("In Maintainance. Please Try Again in 3months");
      // Redirect to the admin page or perform admin-specific actions
      window.location.href = "/handleAdmin";
    } else {
```

```
      alert("Incorrect password. Admin login failed.");
    }
  }

  function loginAsGuest() {
    alert("Guest login successful!");
    // Redirect to the code.html page
    window.location.href = "/aquasentinel";
  }
</script>

</body>
</html>

)=====";

// HTML content for the status page
const char* htmlContent3 = R"=====(
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Status Page</title>
```

```
    <style>
        body {
            text-align: center;
            padding: 50px;
        }

        .status {
            font-size: 24px;
            font-weight: bold;
            margin-bottom: 20px;
        }

        .on {
            color: green;
        }

        .off {
            color: red;
        }
    </style>
</head>
<body>

    <h1>Status Page</h1>
```

```
    <p class="status">Buzzer Status: <span id="buzzerStatus">On</span></p>
    <p class="status">Dam Status: <span id="damStatus">On</span></p>
    <p class="status">Water Storage Status: <span id="waterStorageStatus">On</span></p>
    <p class="status">Agriculture Status: <span id="agricultureStatus">On</span></p>
    <p class="status">Bridge Status: <span id="bridgeStatus">On</span></p>

    <script>
        function updateStatus(statusId, status) {
            var statusElement = document.getElementById(statusId);
            statusElement.innerText = status;
            statusElement.className = status === "On" ? "on" : "off";
        }

        function getStatus() {
            function sendRequest(url, callback) {
                var xhr = new XMLHttpRequest();
                xhr.onreadystatechange = function() {
                    if (xhr.readyState == XMLHttpRequest.DONE) {
                        callback(xhr.responseText);
                    }
                };
                xhr.open("GET", url, true);
```

```
    xhr.send();
}

sendRequest("/getBuzzerStatus", function(response) {
    updateStatus("buzzerStatus", response);
});

sendRequest("/getDamStatus", function(response) {
    updateStatus("damStatus", response);
});

sendRequest("/getWaterStorageStatus", function(response) {
    updateStatus("waterStorageStatus", response);
});

sendRequest("/getAgricultureStatus", function(response) {
    updateStatus("agricultureStatus", response);
});

sendRequest("/getBridgeStatus", function(response) {
    updateStatus("bridgeStatus", response);
});

sendRequest("/getDamDoorStatus", function(response) {
```

```
            updateStatus("damdoorStatus", response);
        });
    }

    // Update status on page load
    getStatus();
    // Update status every 5 seconds (adjust as needed)
    setInterval(getStatus, 1000);
  </script>

</body>
</html>
)=====";

// Handle the root URL login page
void handleRoot() {
  server.send(200, "text/html", htmlContent1);
}

// Handle the guest page
void handleAquasentinel() {
  Serial.println("Handling guest request");
  server.send(200, "text/html", htmlContent3);
}
```

```
// Handle requests to get the status
void handleGetBuzzerStatus() {
  int buzzerState = digitalRead(buzzerPin);
  server.send(200, "text/plain", String(buzzerState == LOW ? "On" : "Off"));
}

void handleGetDamStatus() {
  int damState = digitalRead(damPin);
  server.send(200, "text/plain", String(damState == LOW ? "On" : "Off"));
}

void handleGetWaterStorageStatus() {
  int waterStorageState = digitalRead(waterStoragePin);
  server.send(200, "text/plain", String(waterStorageState == LOW ? "On" : "Off"));
}

// Setup function
void setup() {
  Serial.begin(115200);

  // Set the pin modes for various components
  pinMode(buzzerPin, OUTPUT);
  pinMode(damPin, OUTPUT);
```

```
  pinMode(waterStoragePin, OUTPUT);

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

  // Define routes and handlers for the web server
  server.on("/", HTTP_GET, handleRoot);

  server.on("/aquasentinel", HTTP_GET, handleAquasentinel);

  server.on("/getBuzzerStatus", HTTP_GET, handleGetBuzzerStatus);
  server.on("/getDamStatus", HTTP_GET, handleGetDamStatus);
  server.on("/getWaterStorageStatus", HTTP_GET, handleGetWaterStorageStatus);

  // Start the web server
  server.begin();
  Serial.println("Server started");
```

```
  // Set the initial states of various components
  digitalWrite(buzzerPin, HIGH);
  digitalWrite(damPin, HIGH);
  digitalWrite(waterStoragePin, HIGH);
}

// Loop function
void loop() {
  // Handle incoming client requests
  server.handleClient();
}
```

8.2 Pictures of the system:

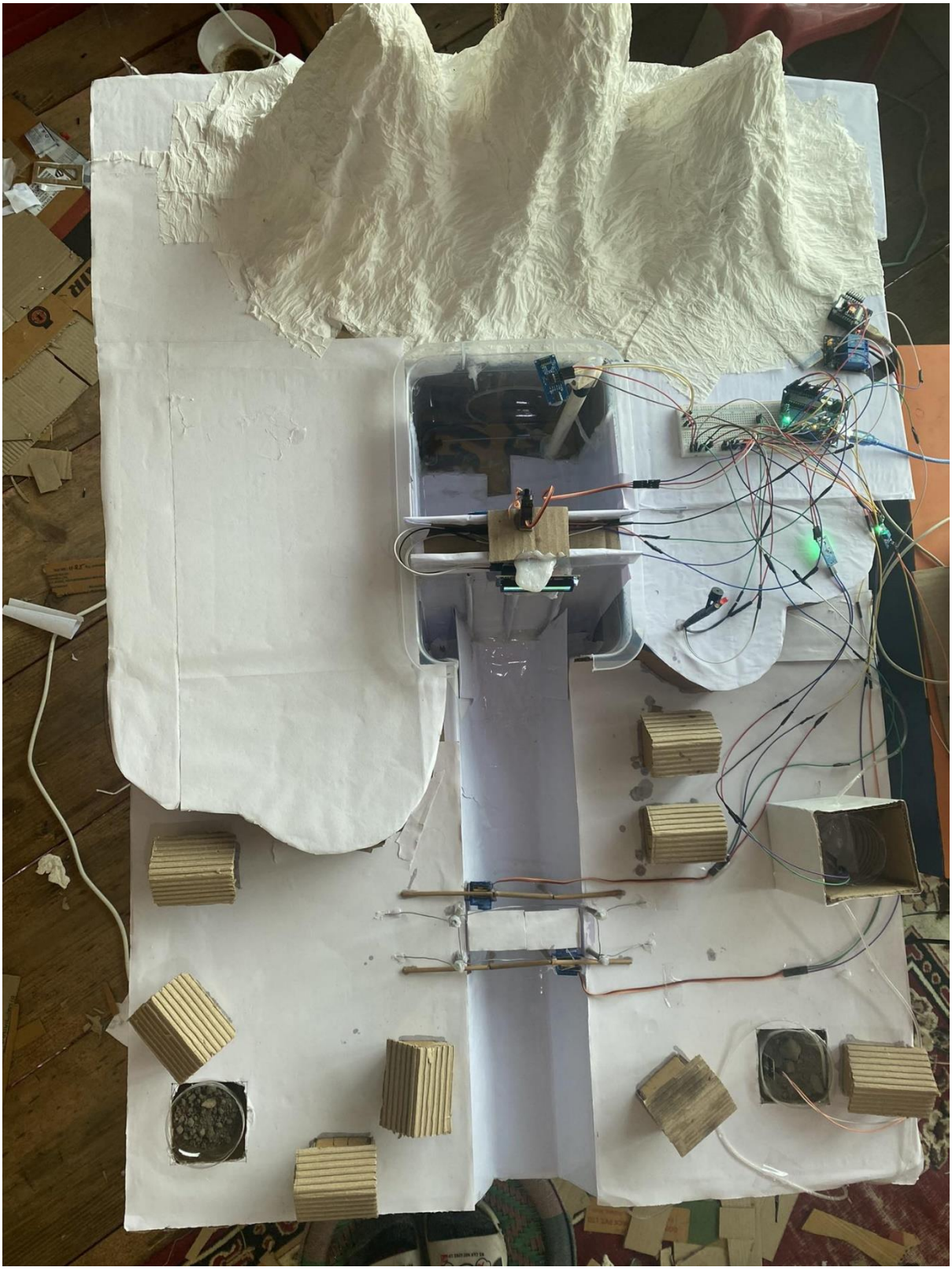

Figure 46: Completed picture of our project.

8.3 Individual contribution plan:

Table 9: Individual Contribution.

Member Name	Task	Description
Rahul G.C.	Project initiation and requirement analysis.	Define the project scope and gather the requirements for system.
Rachit Raj Shrestha	Design and prototype	Design, create and review the prototype.
Prabal Gurung	Development	Develop and review the code for the system.
Prabin Thapa	Testing	Conduct testing of the system and resolve the bugs and issues.
Aman Gurung	Deployment and documentation.	Deploy the system and create the document.

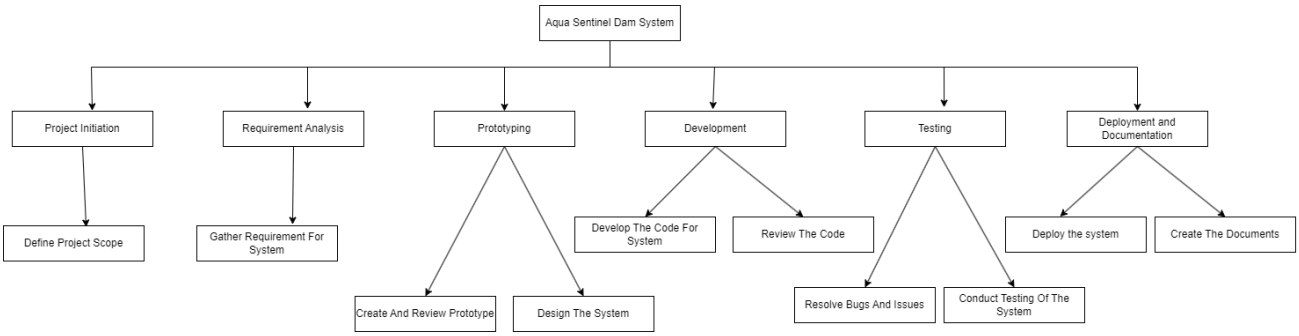

Figure 47: Work break down structure of project.