CS5068NI– Cloud Computing & IoT

50% Group Coursework

Submission: Final Submission

Academic Semester: Spring Semester 2025

Credit: 15 credit semester long module

Title: Water Management and Monitoring and Weathercast

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1	Rohan Prasad Adhikari	NP01NT4A230177	23047505
2	Athit Gurung	NP01NT4A230164	23047490
3	Suraj Shrestha	NP01NT4A230168	23047496
4	Nishan Rai	NP01NT4A230178	23047497
5	Prashan Ghimire	NP01NT4A230046	23047351

GitHub Link:

Assignment Due Date: 15th May, 2025

Assignment Submission Date: 15th May, 2025

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

23047505 Rohan Prasad Adhikari L2N7 Team1.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:95905929

Submission Date

May 14, 2025, 11:46 PM GMT+5:45

Download Date

May 14, 2025, 11:48 PM GMT+5:45

File Name

23047505 Rohan Prasad Adhikari L2N7 Team1.docx

File Size

32.3 KB

29 Pages

5,034 Words

27,646 Characters

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **60** Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 2% Publications
- 11% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **60** Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 2% Publications
- 11% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	University of Newcastle on 2024-12-03	1%
2	Internet	www.coursehero.com	<1%
3	Submitted works	islingtoncollege on 2025-01-23	<1%
4	Submitted works	Leyton Sixth Form College, London on 2024-11-17	<1%
5	Submitted works	St Pauls School London on 2024-03-22	<1%
6	Submitted works	International Islamic University Malaysia on 2017-05-11	<1%
7	Submitted works	Nottingham Trent University on 2025-04-30	<1%
8	Internet	www.instructables.com	<1%
9	Publication	Anudeep Juluru, Shriram K. Vasudevan, T. S. Murugesh. "Let's Get IoT-! - 30 IoT Pr...	<1%
10	Submitted works	Asia Pacific University College of Technology and Innovation (UCTI) on 2023-05-18	<1%

Acknowledgment

We are thankful to Mr. Sugat Man Shakya and Mr. Bishnu Pandey, with their continued support, constructive feedback and constant thrills, they made sure that we were motivated and focused during the entire project development. With their support we were not only able to understand the fundamentals of the theories behind IoT, but also to implement these ideas into practice and actualize these ideas in the form of real-world implementations given the practical work done. Their continuous helps and guidance helped us to creating an enabling learning area presented us with opportunities of discussing such issues as integration of sensor, real-time data monitoring, and microcontroller-based automation with the help of Arduino Uno. Their direction guided us how IoT can address common day-to-day challenges at the areas of smart homes and water conservation two critical areas that contributed much to our final output.

We would like to express thanks to our dear friends whose assistance in making many parts of our project a reality. They came in handy in guiding us in the tuning of accuracy of the ultrasonic sensor, bringing useful bits of advice on breadboard circuit connections and assisting to calibrate the LCD towards live data updates. With their busy schedules, they became available to share practical hands-in experience and work with us to look at issues on the project.

We are genuinely thankful to all those who contributed directly or indirectly to our project success. Due to the encouraging words and the expert technical advice by everyone, our understanding and execution of IoT project became successful with improved version. This was a turning point for us as young tech enthusiasts, and we are in so much gratefulness for all the encouragement and help we got during this journey.

Abstract

Most people residing in residential areas must deal with the dual issues of excessive water supply combined with water tanks overflowing. The inability of people to determine water tank water levels makes users experience water shortages during critical times. Users will fail to identify when the water tank reaches its full capacity despite turning on the pump which leads to excessive water leakage. A water tank monitoring system serves as a solution to manage challenges related to water tanks.

Weather conditions heavily impact daily life because they influence both how people move around and which outdoor tasks they perform. Poor decisions alongside inconvenience occur because of unreliable or delayed weather update systems. The project seeks to build a real-time weather update system through IoT technology which tracks environmental parameters consisting of temperature, in a continuous manner. Weather data collection is performed through sensors connected to cloud infrastructure which processes the information before it distributes the data through mobile applications and web portals to users. The solution helps users receive accurate information with timely delivery in addition to local weather forecast prediction capabilities that improve upon traditional service weaknesses.

It is also possible to use a sensor to monitor the water level, so that if the level falls below a certain level, a notification is sent to the user to turn on the pump. When there is an overflow of water in the water tank, the sensor detects the water level and the pump receives a notification to turn off the pump if the water level rises above a certain level. This system is use to reduce the water waste. A Weather Update IoT Project provides real-time, localized weather data to improve daily decision-making. Traditional forecasts often lack accuracy for specific locations, but IoT sensors monitor temperature, offering precise updates. By delivering accurate and timely weather insights, this system enhances safety, convenience, and resource management in everyday life.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The report is on two crucial IoT-based implementations which handle Water Management and Monitoring along with Weathercast as environmental solutions. The necessity of smart solutions for sustainability grows stronger because water scarcity and climate unpredictability increase. The World Bank reports that global water loss at rates between 30–60% occurs through leakages combined with poor distribution systems primarily affecting urban domains. The Water Management and Monitoring system makes use of IoT sensors alongside cloud platforms and automation for leak detection and powerful water usage monitoring alongside effective water distribution management. Real-time data about weather conditions including temperature along with humidity measurements is available through the Weathercast system to enhance operational forecasting and smart choices in agriculture and transportation together with energy consumption.

1.1. Current Scenario

Population expansion, urbanization, and climate change have made water scarcity and ineffective water management major worldwide issues. Leaks, poor distribution, and a lack of real-time monitoring systems cause significant water waste in many areas. In the same way, weather patterns are become more erratic, impacting daily activities, transportation, and agriculture.

1.2. Problem Statement and Solution

In many places, water management is based on traditional methods that are unable to identify leaks, effectively monitor use, or guarantee fair distribution. Similarly, farmers, commuters, and city planners are impacted by delayed and erroneous weather forecasts, which results in bad decisions about risk reduction, infrastructure protection, and agriculture.

These issues can be resolved by combining automation, and smart sensors in an Internet of Things-based water monitoring and management system. In similar ways, a Weather Update IoT System uses sensor networks to measure temperature, humidity and so on.

1.3. Aim

- To find out how IoT-based weather update and water monitoring and management systems may be used to address the problems of water waste and weather forecasts.

1.4. Objectives

- To figure out how IoT integration may improve decision-making, decrease human intervention, and automate resource management.
- To gain the knowledge about different components used in this project along with Arduino IDE.
- To gain the knowledge and skills about different diagram to make the prototype of the project before jumping directly to build the project.
- To develop an Internet of Things based real-time water monitoring and management system that optimizes usage and reduces waste.
- To develop a localized, real-time weather update system powered by the Internet of Things in order to increase forecasting precision and readiness.
- To analyse the project is complete as per the requirements and should be running successfully.
- To gain the experience about team work and solve the problems while completing this project together without any obstacles.

2. Background

2.1. System Overview

The IoT-based Water Management and Monitoring System performs automatic water tank control along with current weather conditions monitoring in real-time. All system components including Arduino Uno with ultrasonic sensor operate with LCD display and relay module and water pump and weather sensor work to efficiently monitor the tank water level.

An ultrasonic sensor detects the distance between the top of the 10 cm tank and the water surface thus enabling the Arduino to show the current water level on a 16x2 LCD screen. Excess water will activate the pump through the relay control system which will run until the tank reaches 90% capacity. An intelligent feature is added through the weather sensor which detects rainfall alongside particular environmental conditions. The system uses the detected rain condition and displays the message accordingly.

This system involves automatic water level monitoring technology with weather detection capabilities to provide quick and efficient water control with very little human assistance suitable for residential and garden usage and modest agricultural applications.

2.2. Designing Diagrams

This section includes diagrams that shows the project structure through Hardware Architecture drawings and Circuit Diagrams with including Flowchart of the project.

2.2.1. Block Diagram

A block diagram is a representation of a system using simple block that outlines the main components and functions of the system. Each block generally has its own function, component, or procedure and the interconnects between blocks demonstrate the march of data, signals, or control throughout the system. Block diagrams are excellent resources to use in a variety of sectors, such as engineering, electronics, software, or system creation because they can solidify complex systems into concise, intuitive graphics.

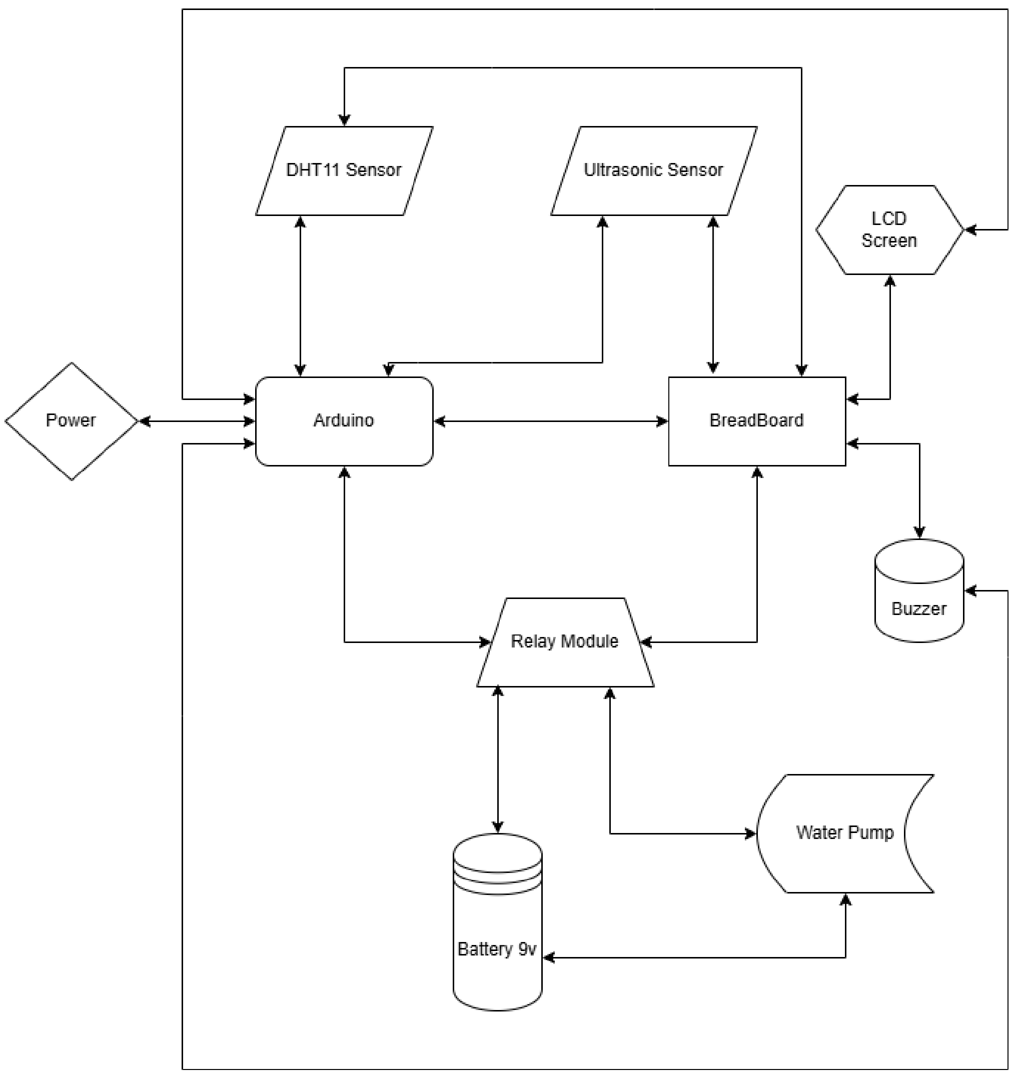

Figure 1 : Block Diagram of the Project

2.2.2. Hardware Architecture

The hardware architecture shows the physical arrangement and wiring structure between all electronic components used for this project. Through this diagram one can understand which components connect to the Arduino microcontroller to complete automation.

The complete hardware diagram was created using cirkitdesigner.com as shown below. The system starts with a central Arduino Uno controller that links up to a breadboard structure for circuit management. An ultrasonic sensor operates as a water level measurement device while the DHT11 sensor performs temperature and humidity monitoring tasks. The relay module operates the water pump following sensor input with taking power source from external battery and the buzzer functions for alert generation. The system displays live system information through the 16x2 I2C LCD display. The connection of all components and power flow through the Arduino relies on jumper wires.

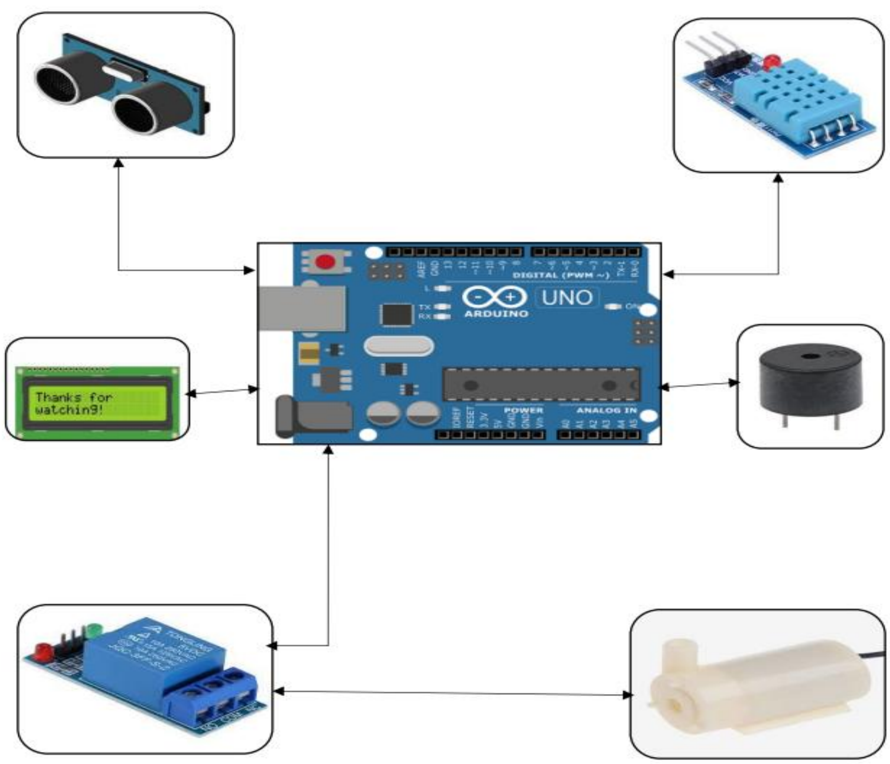

Figure 2 : Hardware Architecture of the Project

2.2.3. Circuit Diagram

A circuit diagram represents electrical circuits visually through graphics and is alternatively known as an electrical diagram or basic diagram or electronic schematic. The circuit components use standardized symbols while solid lines with occasional arrows show the component interconnections. The circuit diagram breaks down language barriers by using a widely recognized set of symbols to represent components such as resistors, capacitors, and transistors, together with solid lines to show the conducting paths, providing a simple and straightforward design. (Tinkercad, 2025)

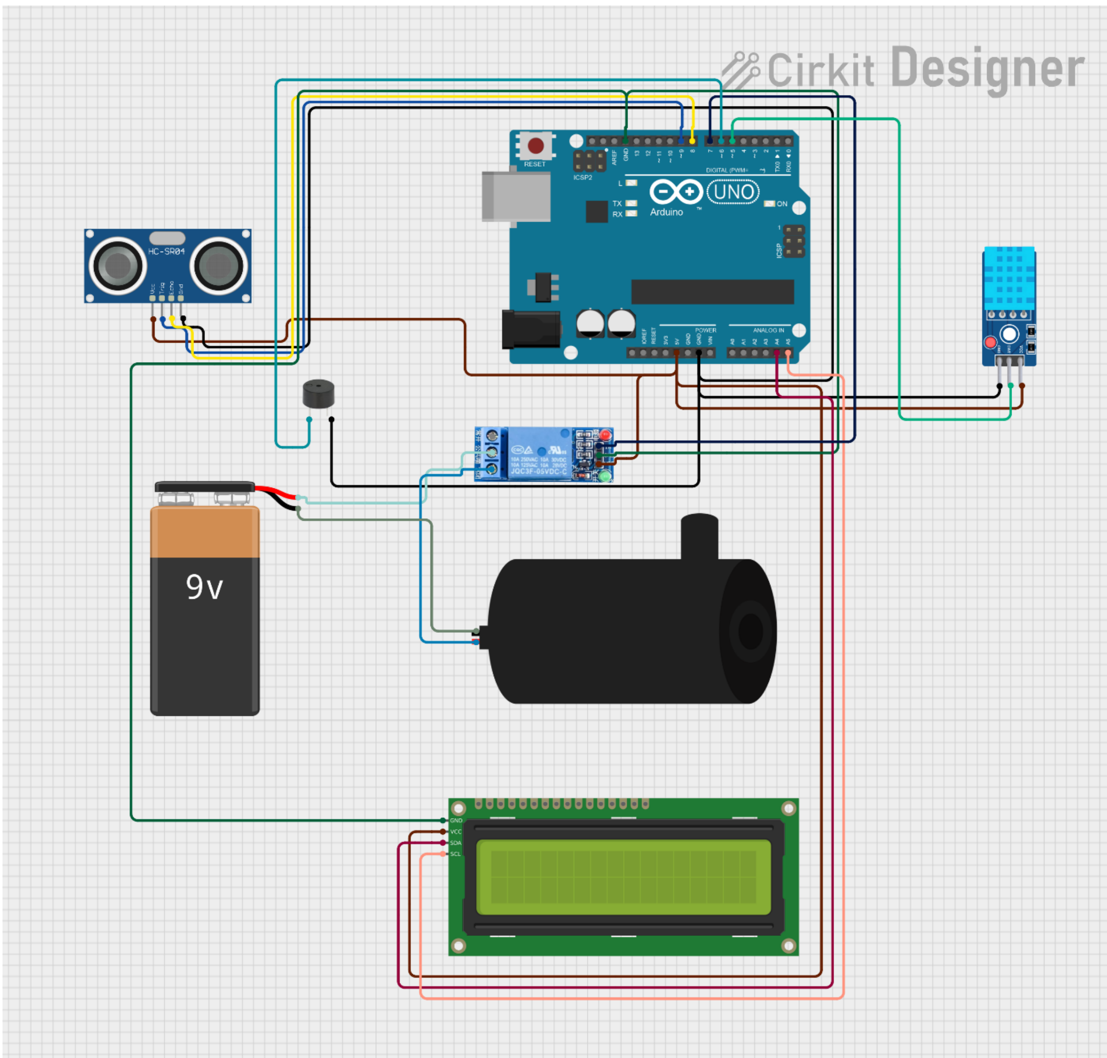

Figure 3 : Circuit Diagram of the Project

2.2.4. Schematic Diagram

A schematic diagram is a diagram that illustrates visually the layout of both the components and the ways things are connected within an electrical or electronic circuit. The layout exhibits the interconnections between components such as sensors, microcontrollers and wires with the help of recognized diagram symbols. It sets forth clarity regarding the arrangement of components in the system and is essential both for its construction and diagnosis of problems.

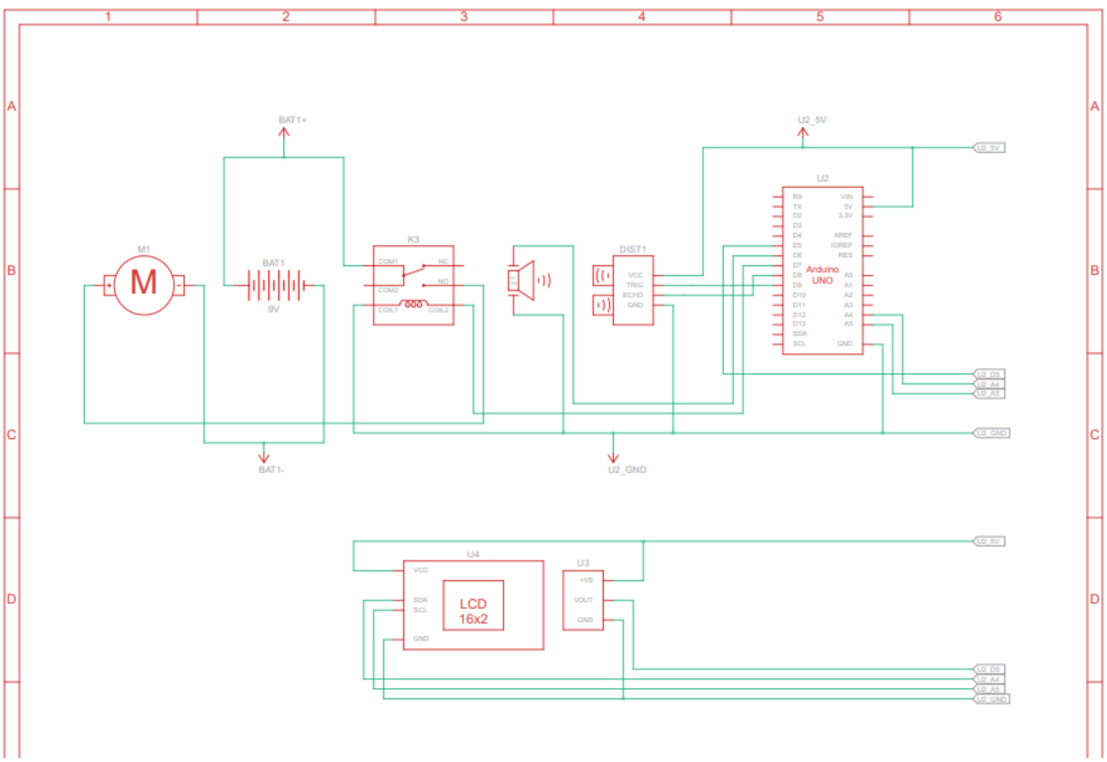

Figure 4 : Schematic Diagram of the Project

2.2.5. Flowchart

Flowcharts represent visual representation of algorithms together with processing decisions that lead to specific goal achievement. A flowchart functions as a step-by-step guide for process analysis and practical implementation for computer programs and manufacturing operations and decision-making processes alike in similar to circuit diagrams for electrical systems. The defined set of arrows connecting symbols leads readers through a logical path which illustrates the control or information flow in various phases. (Draw.io, 2023)

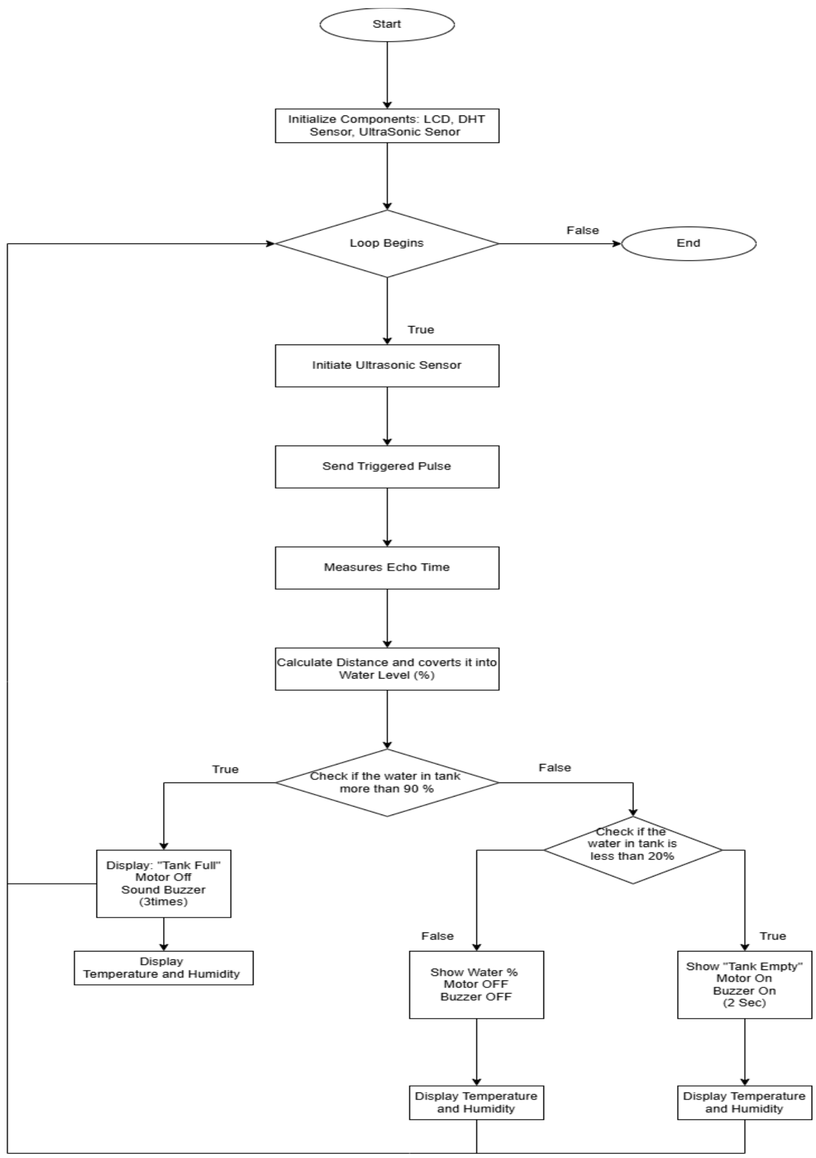

Figure 5 : Flowchart of the Project

2.3. Requirement Analysis

2.3.1. Hardware Components

2.3.1.1. Arduino Uno

The Arduino Uno is the main control unit (Microcontroller) of the system. It reads inputs from the sensors (ultrasonic and DHT11), processes the data using programmed logic, and controls the actuators (relay and pump). It also updates the LCD with the water level, acting as the central coordinator of all operations.

Figure 6 : Arduino Uno

2.3.1.2. Weather Sensor Module (DHT11)

The DHT11 is commonly known as a sensor device designed to track both temperature and humidity levels through digital signal to the Arduino. The DHT11 sensor delivers added value to future system implementations where climate sensitive water usage or data logging becomes necessary. The water level monitoring requirement is not required for the basic setup yet adding environmental sensing through it will increase the project's smart features.

Figure 7 : Weather Sensor Module

2.3.1.3. Breadboard and Jumper Wires

Breadboard and Jumper Wires function as development tools for component connection without soldering it. The breadboard enables straightforward connection for the circuit. Jumper wires are used for connections between sensors, actuators and the Arduino to enable flawless data exchange among all components.

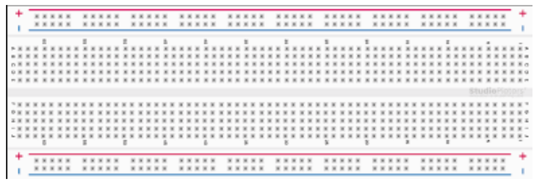

Figure 8: Breadboard

Figure 9: Jumper Wires

2.3.1.4. Ultrasonic Sensor

An ultrasonic sensor worked as a measurement device for determining tank water level distance between surface level and tank height through digital signal to the Arduino. With the time measurement from the Arduino the water level is calculated before being transformed into a percentage value. The sensor monitors the water level of the tank so the system can automatically control the pump operation.

Figure 10: Ultrasonic Sensor

2.3.1.5. Buzzer

It is an output device which plays a notification sound when the tank reaches a low level and when it reaches maximum capacity. An audio warning system can alert users about tank levels regardless of their display viewing status.

Figure 11: Buzzer

2.3.1.6. Relay Module

The relay module functions as an actuator controlled by the Arduino since it behaves as a switch for activating and deactivating high-voltage devices. The Arduino uses a signal to activate the relay system which proceeds to activate the water pump when water levels remain depleted. Through this module the Arduino can activate high-voltage devices using its low-voltage signals in a fix and certain manner.

Figure 12 : Relay Module

2.3.1.7. Water pump

A water pump operates as an actuator through which the tank can receive water below its target level. The relay module directly connects to the water pump throughout its ON/OFF operations control. Operational functionality of the pump occurs only during required the moments when water level is low and the system run efficient and automatic.

Figure 13: Water Pump

2.3.1.8. Power source

The power source activates the water pump without affecting the operating system of the Arduino. The relay module functions as a switch to transmit power that flows from the supply. Relay operation depends on a signal from the Arduino before it completes the circuit to activate the pump when the tank reaches an empty state.

Figure 14: Battery

2.3.1.9. LCD Screen

The LCD display helps to show the current water tank levels through percent indicators in the screen itself. Monitoring your tank status becomes possible through the LCD screen even when mobile apps or cloud connections are unavailable. The implementation of an I2C module simplifies Arduino connections because it reduces the number of required pins.

Figure 15 : LCD Screen

2.3.2. Software

2.3.2.1. Arduino IDE

The Arduino Integrated Development Environment (IDE) serves as program software through which users develop code and transfer it to their Arduino controller boards. Users can write code in C and C++ languages through the IDE which functions on Windows systems and macOS and Linux operating systems (GeeksforGeeks, 2024).

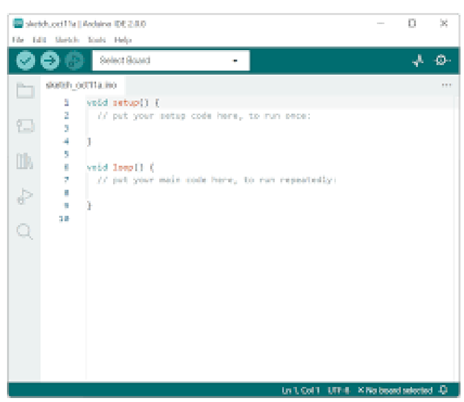

Figure 16 : Arduino IDE

2.3.2.2. Draw.io

Draw.io functions as a free and open-source diagram software to generate flowcharts together with UML diagrams as well as organizational structures and wireframes and additional diagram types. (Draw.io, 2023).

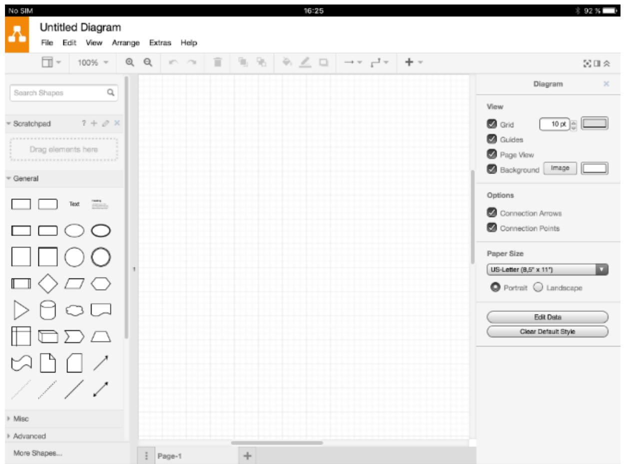

Figure 17 : Draw io

2.3.2.3. Ms Word

Microsoft Word serves as a platform that helps users compose documentation then change or write texts and arrange them before storing it and also for writing papers consisting of letters reports and research papers. (Rodrigo, 2023)

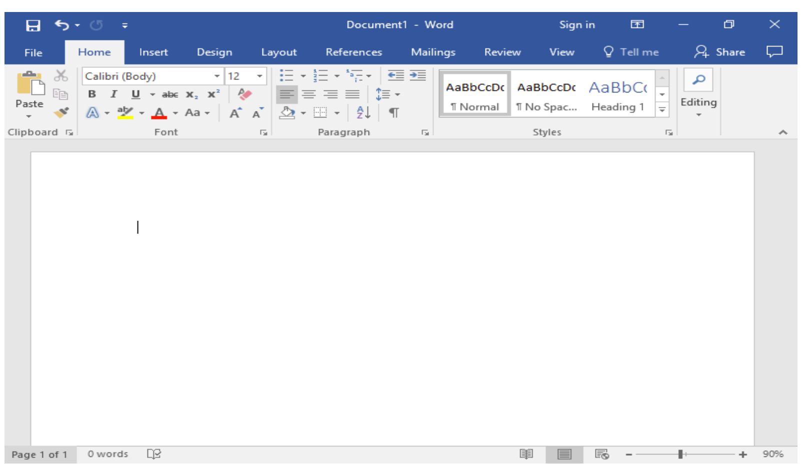

Figure 18 : Ms Word

2.3.2.4. Cirkit Designer

Cirkit Designer is presented as an all-in-one circuit design Integrated Development Environment (IDE) focused on simplifying the process of designing, simulating, and prototyping electronic circuits. (Cirkit Designer , 2025)

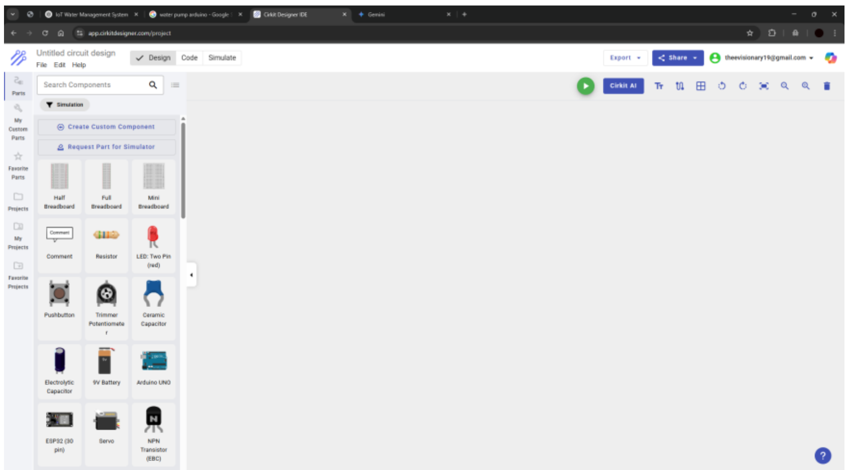

Figure 19 : Cirkit Designer

2.3.2.5. Tinkercad

Tinkercad is a free, user-friendly, web-based CAD (Computer-Aided Design) software developed by Autodesk. Launched in 2011, it has become a popular platform, especially in education, for introducing beginners to 3D design, electronics, and coding. (Tinkercad, 2025)

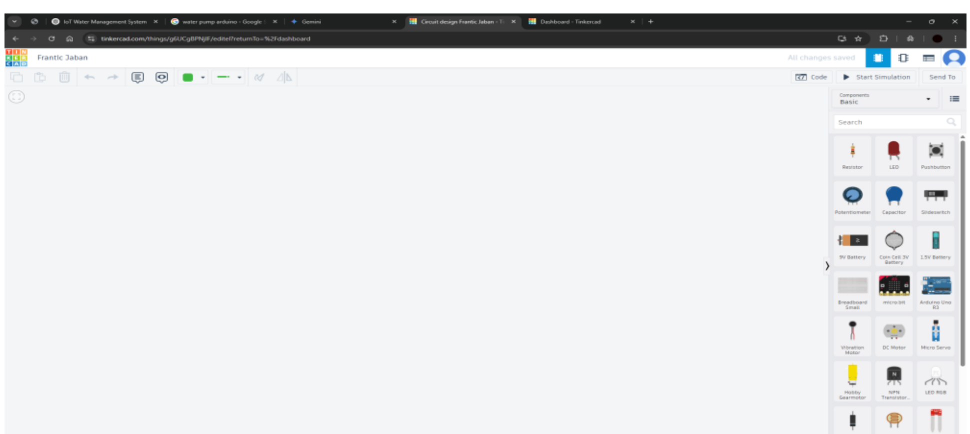

Figure 20 : Tinkercad

3. Development

In this portion of the report, the project's development process is described in detail. Here is a summary of the project's overall state from planning to completion.

3.1. Planning and Design

This project focused on developing an IoT-based Water Tank Monitoring and Management System to handle frequent household problems with water overflow and scarcity. The project functions through dual purposes by tracking water level in the tank and conducting automatic water management using smart sensors and controls. The development process required a detailed knowledge and strategic planning because such technical projects need proper foundation before beginning actual work.

The project started by planning to make a basic system which would show water levels through an ultrasonic sensor connected to LCD display technology. During planning phases, we decided to expand the project by including automatic water pump control and weather-based decision-making which evolved the system into an enhanced practical solution. The addition of these features enhanced project value significantly for families who wanted to save human efforts and prevent water loss.

The idea of this project was clicked due to everyday challenges that include both pump overflow from leaving it running and running out of water at many times. This system functions as a simple technical design because it tackles real-life issues related to tank overflow and empty reserves. The project came to existence by employing multiple electronic components including sensors, relays, a buzzer as well as a microcontroller. The components used in the project is explained in a detailed form about the devices and their functions.

3.2. Resource Collection

The project required multiple devices and tools for successful development and functionality demonstration. Different locations were selected based on availability and required needs to obtain these components. The resource department of Islington College managed about eighty percent of essential system components to support basic hardware application. The components obtained from the resource department at Islington college are listed below:

- Arduino
- Jumper Wires
- Ultrasonic Sensor
- Breadboard
- Relay Module
- Buzzer
- LCD Screen
- Water Pump

The remaining components, including Weather Sensor (DHT11), Jumper Wires, Batteries and supporting materials, were arranged personally by the group members. Each member actively contributed to sourcing, and purchasing the necessary items to keep the project on track to the completion.

3.3. System Development

The system functions as an IoT based Water monitoring and management system alongside weather forecasting capabilities to prevent both water overflow and water shortage risks. The project uses different devices to prevent water-related problems through monitoring and management systems which minimize and reduce water overflows in the different application areas. The system development is categorized in four different phases which are described in detailed form and that are listed below:

3.3.1. Phase 1: Connecting Arduino Uno with Laptop and Breadboard

This is the first phase of the project which aims to focus on important foundation for the project. The objective was to establish a simple interface for connecting Arduino Uno to laptop which allows to upload code and have serial communication easily. A breadboard was used as a major component in system to organization and also for circuit connections with different components.

The Arduino Uno used as the **central microcontroller** in this project. It receives information from sensors, and sends signals to the devices that should take action accordingly. Before any external components were connected within, it was very important to ensure that there was a stable connection between the Arduino and the laptop for code uploading, serial monitoring and also for power source.

The use of breadboard was equally important at this stage because it allowed to run prototype and tune circuits, easily attach or detach equipment and make sure that could test design without getting into soldering pins of the components. Such flexibility was very precious during the debugging phase, enabling to adjust easily without getting into permanent connections.

- The Arduino Uno was connected to the laptop using a **USB A to B cable**, which allowed to upload code and also provided power during initial testing phases.
- A **breadboard** was placed on the working surface to act as the prototyping base which allowed to have easy connection with the help of jumper wires and components without soldering.

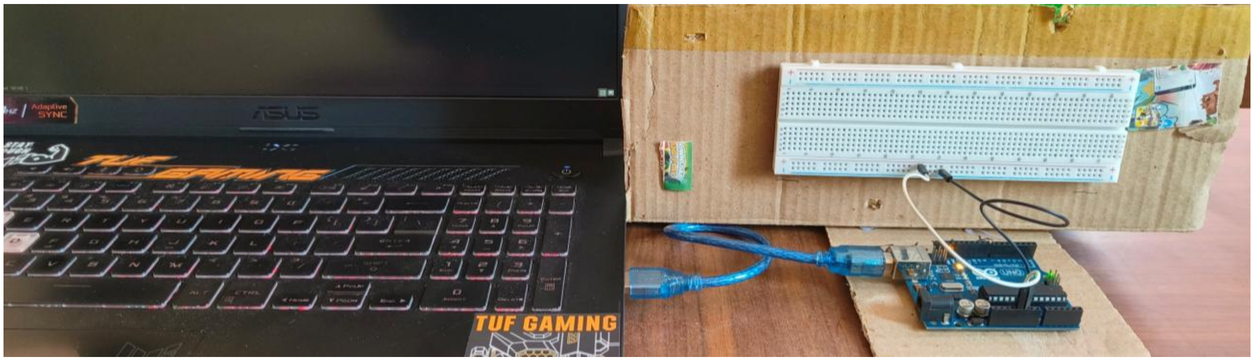

Figure 21: Connecting Arduino Uno with Laptop & Breadboard

3.3.2. Phase 2: Connecting Sensor and Input devices with Arduino

In this phase, mainly focused upon preparation and connection of the input devices that provide Realtime data into the system. **Ultrasonic Sensor is used** for monitoring water level and **DHT11 Sensor** is used to measure weather conditions in a real time.

Figure 22 : Connecting all Input Devices

The **Ultrasonic Sensor (HC-SR04)** was used in this project for measuring distance using sound waves. The ultrasonic sensor was placed on the top of the water tank to measure the distance from the water surface present in the tank. From this information, the Arduino ensures that the tank is full or it needs more water to full the tank.

- The Ultrasonic Sensor (HC-SR04) was configured to measure the water level by identifying distance from the water tank. The VCC pin of the ultra-sonic sensor connected to the 5V pin on the Arduino, and the GND of the ultra-sonic sensor connected to the GND of the Arduino to provide power. The Trig pin was connected to digital pin 9 and the Echo pin was connected to digital pin 10 for the communication between the Arduino to send and receive ultrasonic pulses to calculate and measure distance.

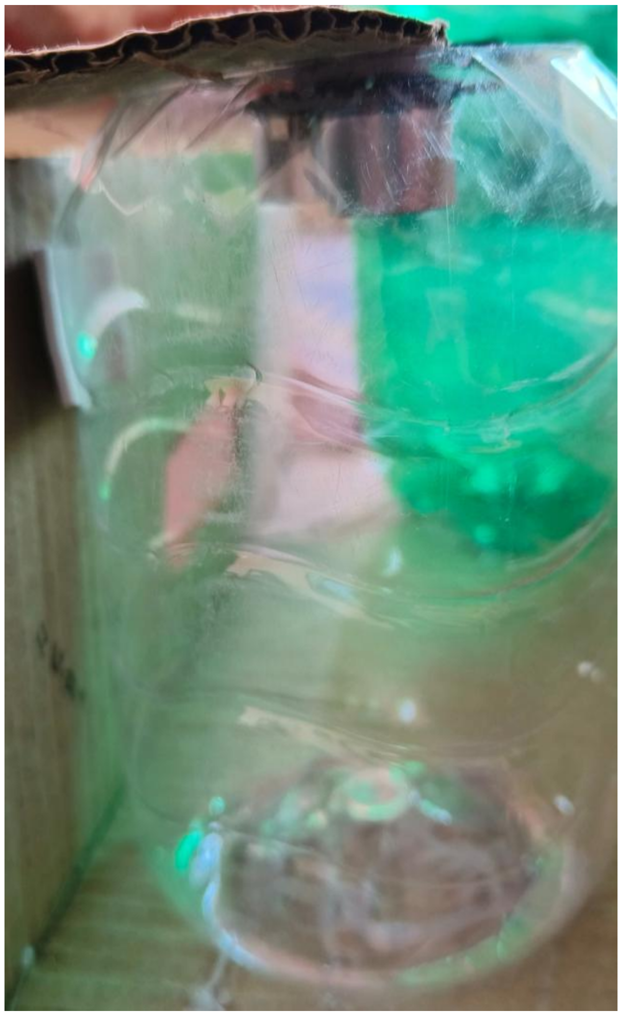

Figure 23 : Ultrasonic Sensor connected

A DHT11 Sensor was added to the system, which is a digital sensor and its ability to measure both temperature and humidities. Through this sensor, system acquires the ability to monitor weather conditions thereby making it more useful in real-time applications such as smart homes and also helps in agriculture.

- A DHT 11 Sensor was used to monitor surrounding temperature and humidity levels, helping the device to further information of the environment. For the DHT11 weather sensor, its VCC pin connected to the 5V pin on the Arduino, GND connected to GND, and its data pin to digital pin 5. It allowed the Arduino to collect environmental data from the sensor using the DHT.h library installed in the Arduino IDE.

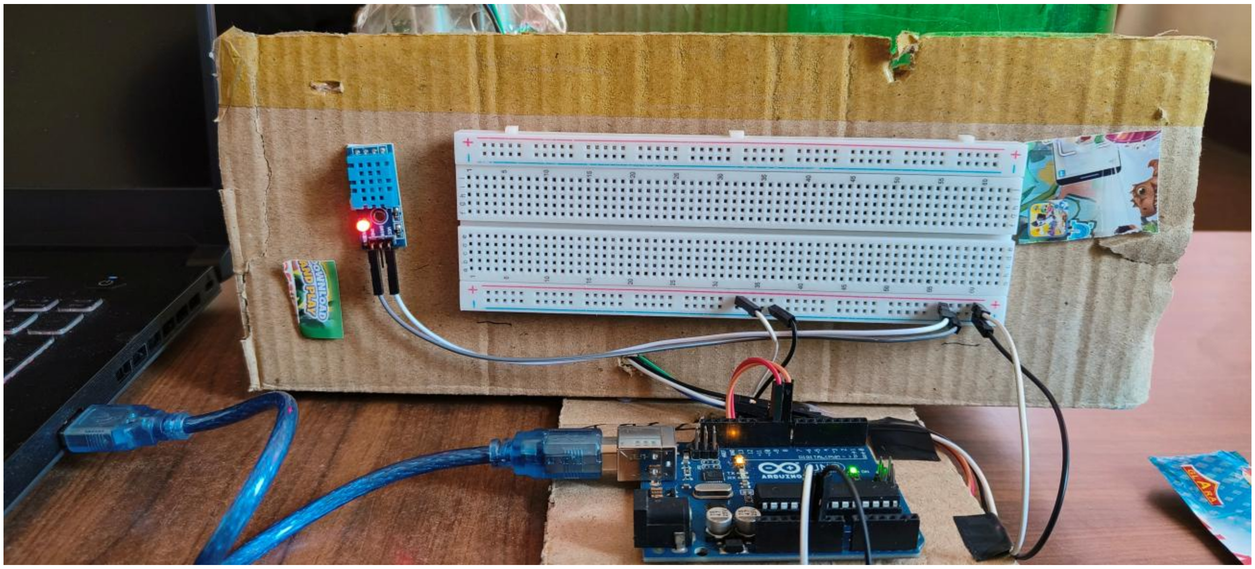

Figure 24 : DHT11 Sensor Connected

Once raw data are received by the Arduino from these sensors, it assesses the information to performs actions like starting the pump, displaying stats on the screen and sound alerts through buzzer.

- Both the sensors are input devices which provided a feed with real-time environmental data to the Arduino for analysis also to perform action accordingly.

3.3.3. Phase 3: Connecting LCD Screen and Output devices with Arduino

This is the third phase of the system development which was focused on the to connect the output and control devices, so they could display system status and execute actions based on the received sensor values. With this integration, the system achieves responsiveness through the provision of feedback and automation.

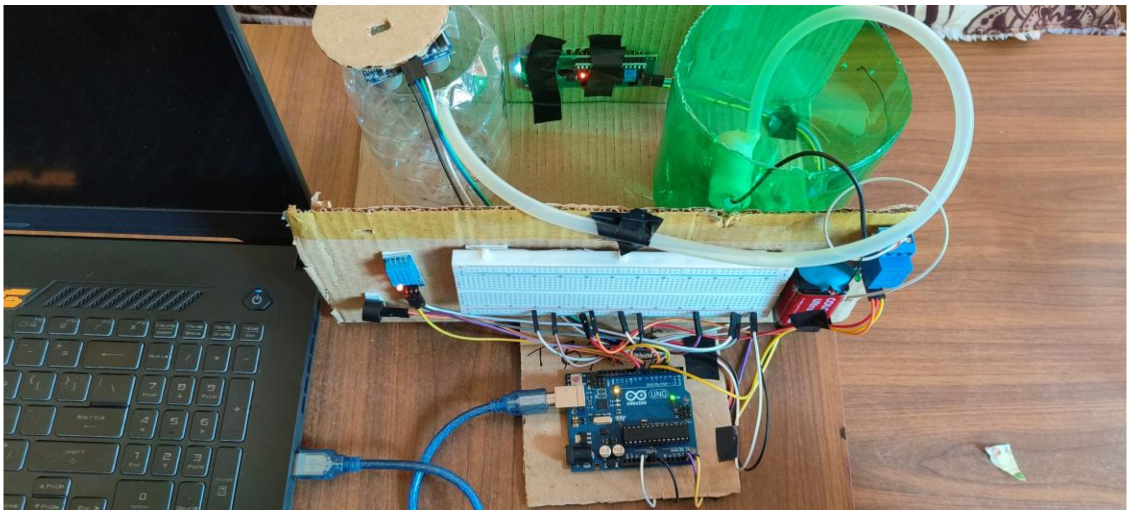

Figure 25 : Connecting all Output Devices

A 16x2 LCD Screen was used to display the current levels of water, temperature, and humidity. This LCD Screen (16x2) is the primary means for the user to monitor important values as they happen. A potentiometer is used to manage the contrast manually, ensuring improved visibility, while an I2C module was added to make connections easier and reduce digital pin usage when required.

- The LCD Screen was used to show the water level, temperature, and humidity. By adding an I2C module to a 16x2 LCD display, connecting the device was made much easier. The I2C module’s VCC connected to 5V on the Arduino, and ground (GND) was connected to the Arduino’s ground (GND). The SDA connected to pin A4, and the SCL connected to pin A5 on the Arduino for proper communication. With the help of these configuration, the LCD displays real-time sensor details using data connections.

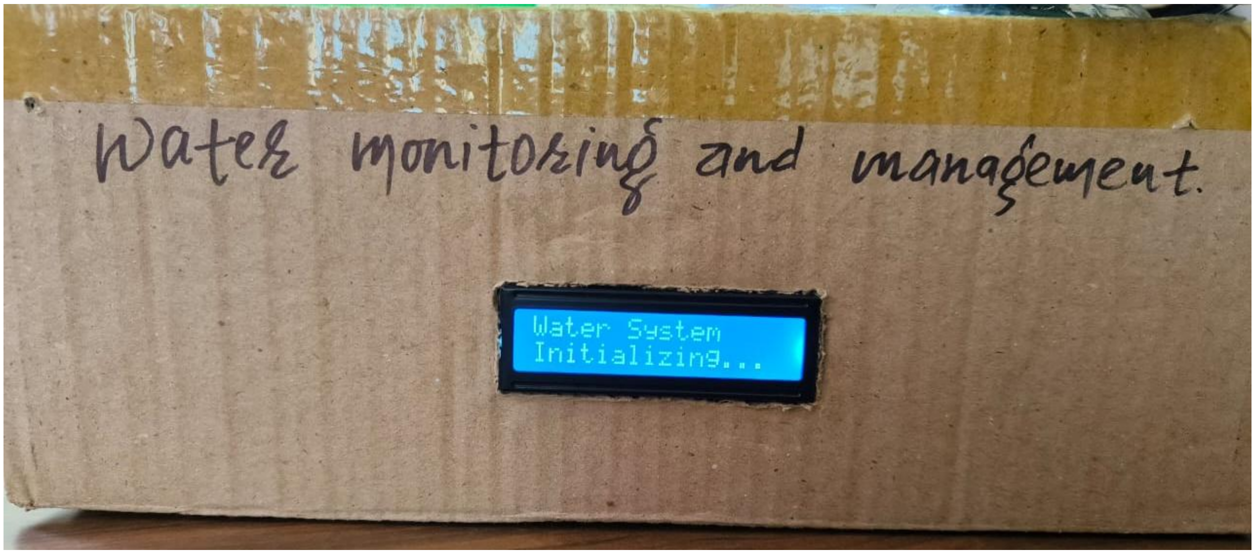

Figure 26 : LCD Connected

A buzzer was used to have an audio alert system integrated in this project. A sound from the buzzer alerts users whether the water is running low or the tank is already full, ensuring they get notice even if they are not paying attention to the LCD screen.

- A buzzer was connected to the system to alert users if the tank was almost empty or full. A buzzer was used to send sound alerts to the user. The positive side of the buzzer was attached to digital pin 7 on the Arduino, and another leg was connected to GND. The buzzer was configured to have an alarm alert when the water tank reached its highest or lowest level.

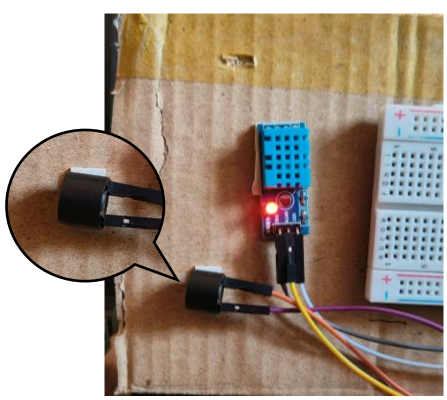

Figure 27 : Buzzer Connected

To the control the water pump, a relay module was included to work as the connection between the Arduino and the water pump. As a result, the relay module helps the Arduino to control the pump by turning it ON or OFF whenever necessary by reading the data from the sensors used.

- A relay module is used to digitally operate and manage the water pump through it. The relay's VCC was connected to the Arduino's 5V, GND to the Arduino's GND, and IN was connected to digital pin 6. It helps the Arduino to manage the relay module with a digital output and through relay module water pump too.

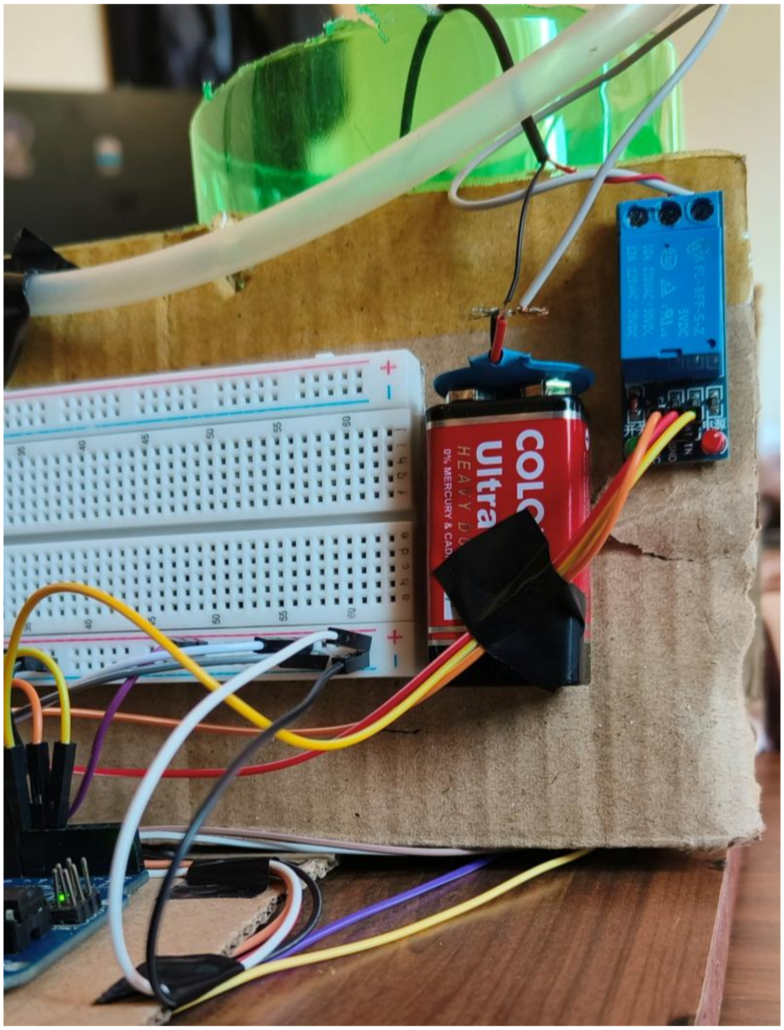

Figure 28 : Relay Module Connected

The water pump is an output actuator which is responsible for operating and fills the tank whenever its level is nearly empty. As soon as the tank reaches 90% full state, the relay module ensures that the water pump is switched OFF to keep the tank from overflowing of water.

- The relay module worked as the pathway for the water pump to perform the task of tank to fill it automatically. The water pump was connected to the relay module using the NO and COM output pins. The relay module performed as a connection between the Arduino and the 9V power supply to start or stop the pump. The pump receives signal from the Arduino whenever the water level was low, and turned off automatically once the tank was approximately 90% of the tank.

Figure 29 : Water Pump Connected

3.3.4. Phase 4: Writing, compiling and uploading code in the Arduino

This is the final and most crucial phase of the project which involved Software development and overall logical control of the system. All communication between the input-output devices of system was managed by code written in the Arduino IDE using C/C++ **programming**.

The code is responsible for handling all decision-making needs of the system. The code receives data from the sensors, carries out necessary calculations, shows the results on the LCD, and gives digital commands to the relay and buzzer. Programming logic was used to define when the pump would start or stop, according to the instruction received by the Arduino uno.

Once the code was written and tested, it was compiled and uploaded to the Arduino Uno from Arduino IDE.

- The system logic was programmed using the **Arduino IDE**, where the code is written and compiled the in **C/C++**.
- After final verification, the code was uploaded onto the Arduino Uno via the same USB connection and real-time testing was done to verify the entire system's performance.

Figure 30 : Code Compiled Successfully

Figure 31 : Code Uploaded Successfully

4. Results and Findings

4.1. Result

The concept for creating this IoT-based Water Tank Monitoring and Management System was already in mind before development began because its potential benefits were easy to predict. The objective focused on solving regular water quality issues by implementing automatic controls in combination with real-time tracking methods.

A combination of ultrasonic sensor and DHT11 weather sensor together with LCD display and relay module and buzzer and water pump achieved successful integration as the final product in development. The buzzer function started playing when the tank reached either an empty state or approached its full capacity level. The device operated its motor automatically according to water tank levels and presented weather data including temperature and humidity through LCD screens. All test scenarios demonstrated the correct functionality of system components within their specified operational criteria.

The project finished with a working prototype which automatically controlled the water tank operations successfully. The device solved the problems of water leakages and insufficient supply in an efficient way that can be used by everyday households.

4.2. Findings

This section demonstrates the successful operation of the project through multiple tests which resulted in 100% successful rate.

4.2.1. Test – 1

Test No:	1
Objective:	To show the execution of the code.
Action:	The code in the Arduino IDE must be verified and will have to upload the code in the Arduino Uno with the help Arduino IDE.
Expected Result:	While compiling the code, it should compile successfully without any errors.
Actual Result:	The code compiled successfully without any error.
Conclusion	Test was Successful.

Table 1 : To show the execution of the code

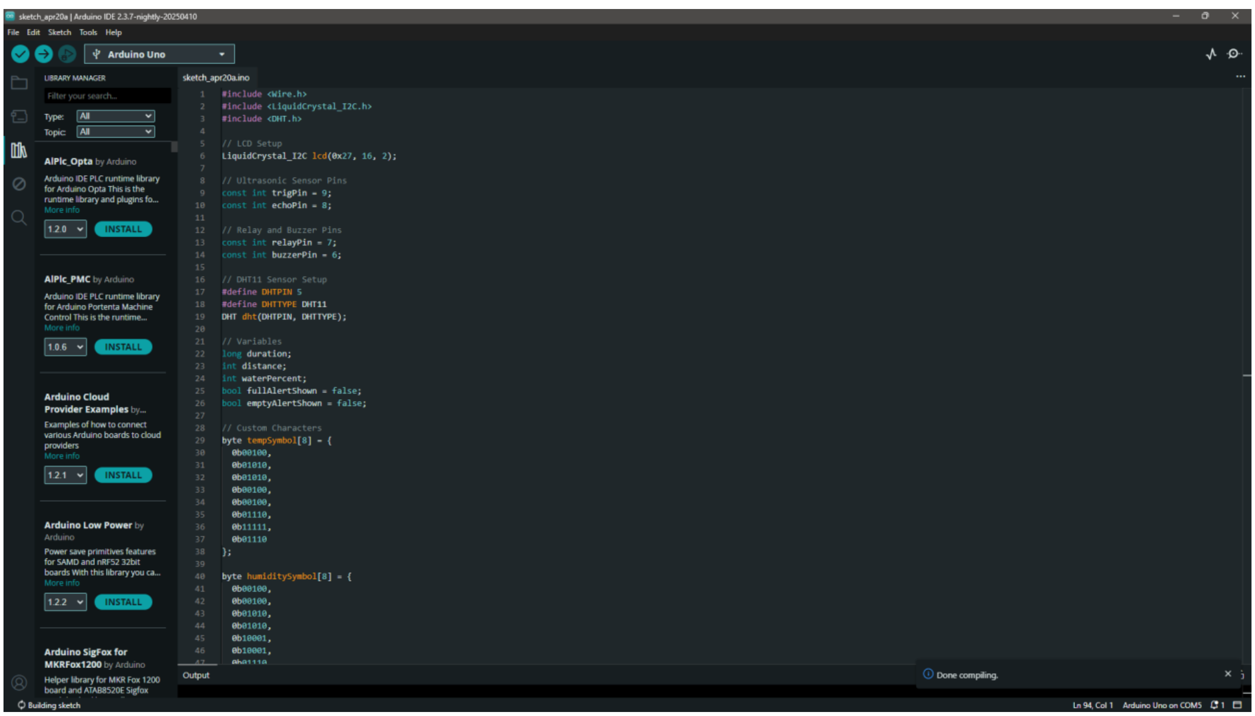

Figure 32 : Code Compiled Successfully

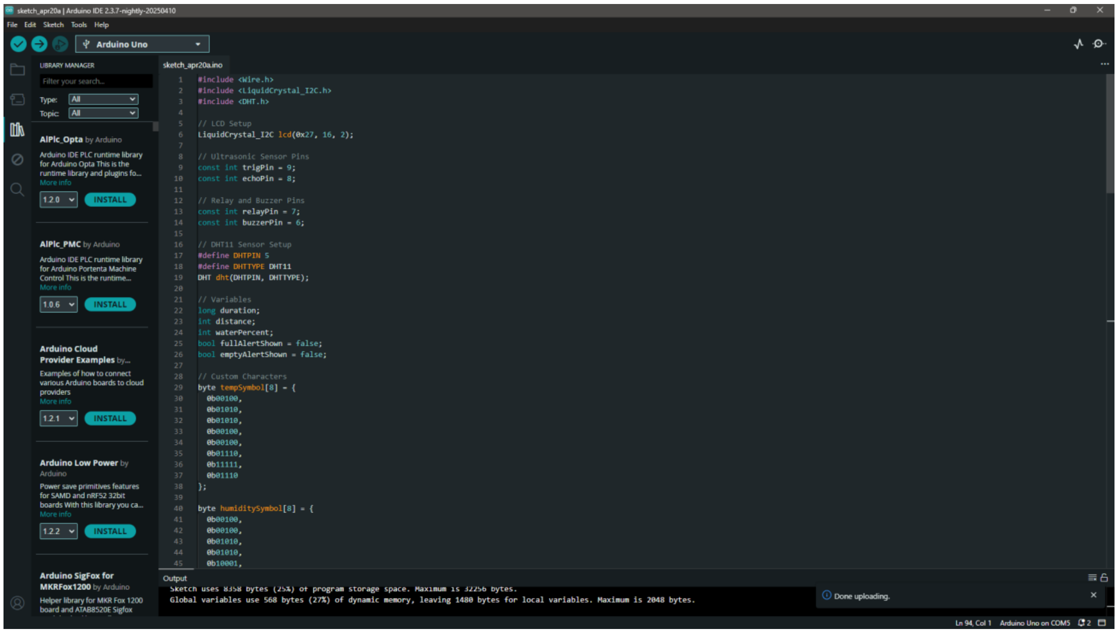

Figure 33 : Code uploaded successfully to the Uno

4.2.2. Test – 2

Test No:	2
Objective:	To test the accuracy and responsiveness of the ultrasonic sensor in measuring water level.
Action:	Place an object at different distances from the sensor and observe the readings on the LCD.
Expected Result:	The LCD should display water level percentage based on object distance.
Actual Result:	The LCD displayed water level accurately and updated in real-time with distance changes.
Conclusion	The ultrasonic sensor is working correctly and provides reliable water level measurements.

Table 2 : To test the accuracy and responsiveness of the ultrasonic sensor in measuring water level.

Figure 34 : Water level measured through Ultrasonic sensor i

Figure 35 : Water Level measured through Ultrasonic sensor ii

Figure 36 : Water Level measured through Ultrasonic sensor iii

Figure 37 : Water Level measured through Ultrasonic sensor iv

4.2.3. Test – 3

Test No:	3
Objective:	To verify if the water pump is physically pumping water when activated.
Action:	Simulate a low water level condition and observe whether water flows from the pump.
Expected Result:	Water should start flowing when the pump is turned ON. water should not flow when the pump is OFF.
Actual Result:	Water flowed correctly when the pump was activated and stopped when deactivated.
Conclusion	The water pump is working properly and is capable of pumping water as expected.

Table 3 : To verify the relay module switches the water pump on and off based on water level.

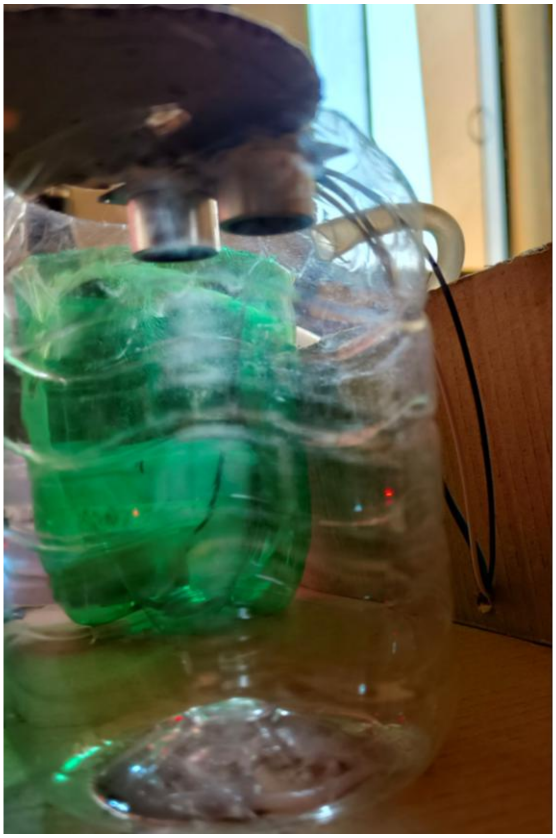

Figure 38 : Water pump is off and not flowing water

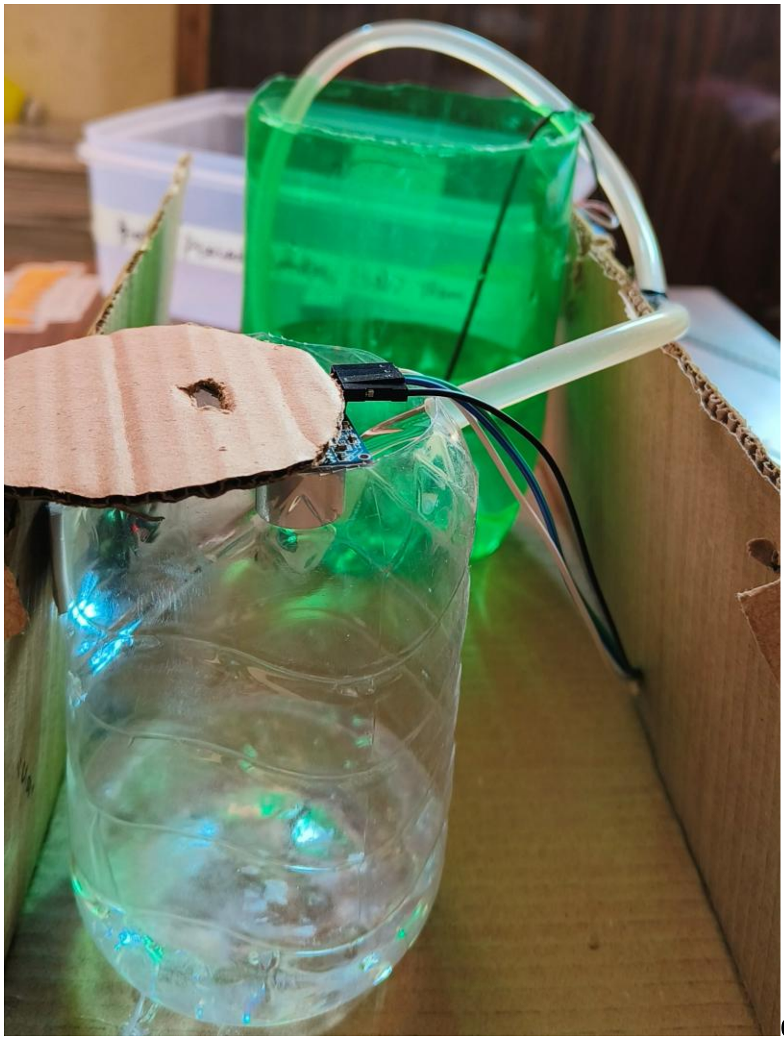
q

Figure 39 : Water pump is on and water flows

4.2.4. Test – 4

Test No:	4
Objective:	To verify if the DHT11 sensor correctly measures and displays temperature and humidity.
Action:	Power the system and observe the LCD for real-time temperature and humidity readings.
Expected Result:	The LCD should show accurate temperature in °C and humidity in %.
Actual Result:	The LCD displayed changing temperature and humidity values correctly
Conclusion	The DHT11 sensor is functioning properly and responds to environmental changes.

Table 4 : To verify if the DHT11 sensor correctly measures and displays temperature and humidity.

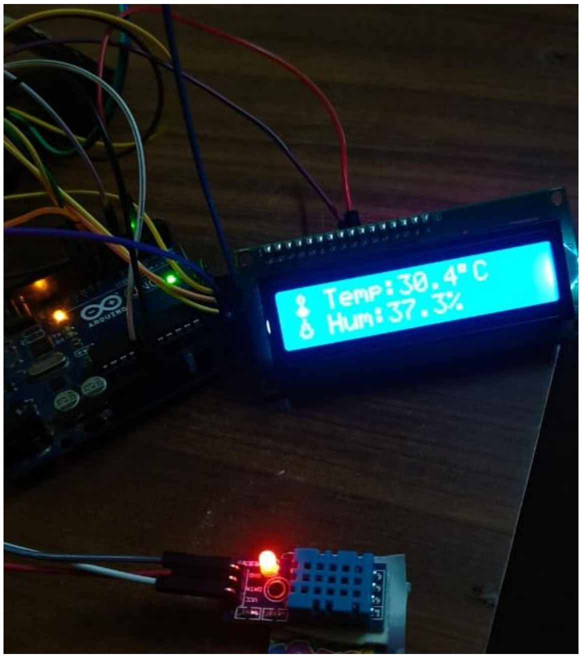

Figure 40 : Temperature reading through DHT11 Sensor i

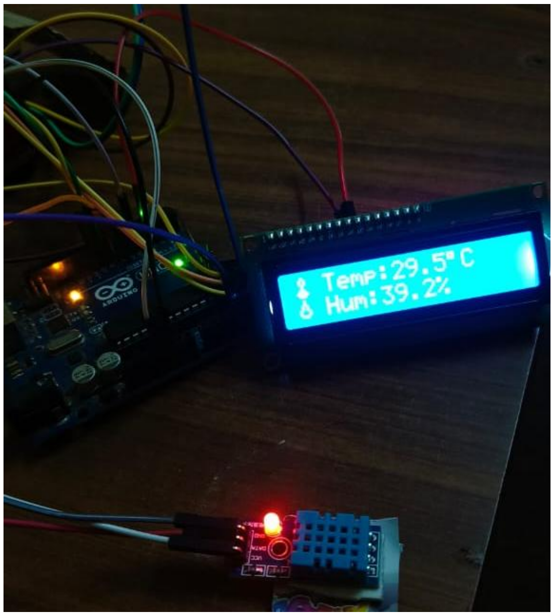

Figure 41 : Temperature reading through DHT11 Sensor ii

4.2.5. Test – 5

Test No:	5
Objective:	To identify the issue of weather info showing "Err" when the pump is running.
Action:	Observe the LCD display while the pump is turned on and check the temperature and humidity readings.
Expected Result:	The weather information should display correctly even when the pump is running.
Actual Result:	Sometimes, the LCD shows "Err" for temperature and humidity when the pump is on.
Conclusion	Interference or power fluctuations caused by the pump affect DHT11 readings.

Table 5 : To identify the issue of weather info showing "Err" when the pump is running.

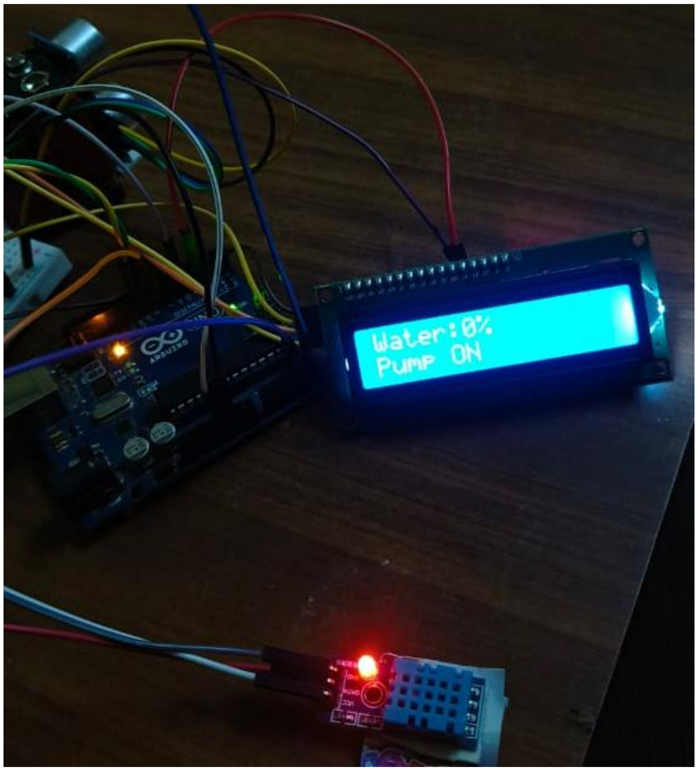

Figure 42 : Error shows when pump is on

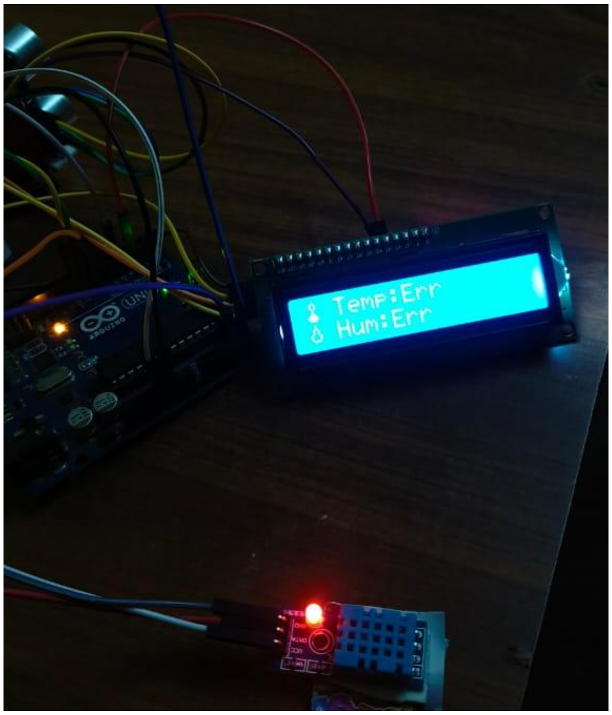

Figure 43 : Error shows when pump is on and power got fluctuated

5. Future Work

The water level monitoring and management system possesses substantial potential to advance in future development. The present version serves as an initial basic prototype yet requires multiple improvements to shift it from concept stage to practical implementation. Research into future development should focus on producing a modified version for installing in domestic and industrial water storage tanks. The system's expansion capabilities along with durability enhancements would enable deployment in managing water consumption across bigger platforms. The system can be remotely monitored through smartphone or web interface after addition of Wi-Fi functionality or other IoT communication protocols. Such integration would let users have enhanced control and convenience over their water delivery management.

Another effective advanced enhancement would be the addition of weather forecasting capabilities. The system generates better water usage decisions because it evaluates current weather conditions with real-time data. The system could delay tank refilling once rainfall forecasts appear which enables efficient water conservation while decreasing wastes. Moreover, combining sensor data with weather analytics could open the door for predictive water management, where the system learns usage patterns and environmental conditions to make intelligent decisions automatically. The upcoming developments will enhance user satisfaction as well as water efficiency and promote sustainable resource management systems.

6. Conclusion

This IoT-based Water Monitoring and Management System, combined with live weather information, helped us to learn about how advanced technology solves daily challenges. The system effectively showed that sensors, automation, and microcontroller programming could resolve problems such as tank overflows, water loss, and unawareness about environmental conditions. The use of an ultrasonic sensor helped us in real-time monitoring of water level and automated water pump operation through a relay module. Therefore, there was no unnecessary water wastage, since the pump automatically shut off once the tank reaches the required level, and manual tasks were also reduced through this project. Furthermore, the system became more adaptable for home or agricultural purposes due to the addition of a DHT11 sensor, which helps in live temperature and humidity information.

This project provided an excellent opportunity to learn about the cooperation between input and output devices through the Arduino Uno. The combination of the LCD screen for display, which displays the real-time information, and the buzzer in giving sound alerts was fundamental to building a functioning smart water system. Real-life technological problems including sensor calibration, power disturbances, and display errors were identified and addressed during the project. Encountering and resolving these issues contributed to stronger debugging skills, better logic formation, and improved understanding of hardware use. Creating clean and efficient programs in the Arduino IDE allowed a practical introduction to microcontroller programming, a basic skill for IoT systems.

This project taught a lesson in the importance of teamwork, and critical thinking. Everyone in the group played an active part from gathering the resources through to the final step, taking part in implementing the coding, constructing the hardware, and solving any technical issues that came up. This system could be made more useful and reach a more people by implementing remote control through Wi-Fi, automatic rain sensing, or cloud data storage are added. In conclusion, the project not only helped to meet academic goals but also inspired to think about how technology can be used to support sustainability and improve lives through smart automation.

7. References

Cirkit Designer , 2025. *Design, Simulate and Prototype Circuits With Ease.* [Online]
Available at: https://www.cirkitstudio.com/
[Accessed 5 May 2025].

Draw.io, 2023. *Use the draw.io editor.* [Online]
Available at: https://www.drawio.com/doc/getting-started-editor
[Accessed 06 April 2025].

GeeksforGeeks, 2024. *Arduino Integrated Development Environment (IDE) v1.* [Online]
Available at: https://www.geeksforgeeks.org/arduino-integrated-development-environment-ide-v1/
[Accessed 06 April 2025].

Rodrigo, A. L., 2023. *What Is Microsoft Word (Definition)? What Is MS Word Used For? (+Top Features).* [Online]
Available at: https://business.tutsplus.com/tutorials/what-is-microsoft-word-definition--cms-34990
[Accessed 06 April 2025].

Tinkercad, 2025. *Tinkercad.* [Online]
Available at: https://www.tinkercad.com/
[Accessed 5 May 2025].

8. Appendix

8.1. Individual Contribution plan

Name	Task
Athit Gurung	Designing Diagrams and Background Portion.
Nishan Rai	Documentation, Formatting, and Introduction Portion.
Rohan Prasad Adhikari	Writing Scripts and Programming and Testing.
Prashan Ghimire	Device Deployment and Future works.
Suraj Shrestha	Debugging and Testing.

Table 6: Individual Contribution plan

8.2. Code

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <DHT.h>


// LCD Setup

LiquidCrystal_I2C lcd(0x27, 16, 2);


// Ultrasonic Sensor Pins

const int trigPin = 9;

const int echoPin = 8;


// Relay and Buzzer Pins

const int relayPin = 7;

const int buzzerPin = 6;


// DHT11 Sensor Setup
```

```
#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Variables
long duration;
int distance;
int waterPercent;
bool fullAlertShown = false;
bool emptyAlertShown = false;

// Custom Characters
byte tempSymbol[8] = {
  0b00100,
  0b01010,
  0b01010,
  0b00100,
  0b00100,
  0b01110,
  0b11111,
  0b01110
};

byte humiditySymbol[8] = {
  0b00100,
  0b00100,
  0b01010,
  0b01010,
```

```
  0b10001,
  0b10001,
  0b01110,
  0b00000
};

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(relayPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  lcd.begin(16, 2);
  lcd.backlight();

  dht.begin();

  // Create custom characters
  lcd.createChar(0, tempSymbol);      // Address 0 for temperature
  lcd.createChar(1, humiditySymbol);  // Address 1 for humidity

  digitalWrite(relayPin, HIGH);  // Pump OFF initially
  digitalWrite(buzzerPin, LOW);

  lcd.setCursor(0, 0);
  lcd.print("Water System");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
```

```
  delay(2000);
  lcd.clear();
}

void loop() {
  // Read Ultrasonic Sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  waterPercent = map(distance, 2, 10, 100, 0);
  waterPercent = constrain(waterPercent, 0, 100);

  // Read Temperature and Humidity
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (waterPercent < 90) fullAlertShown = false;
  if (waterPercent > 0) emptyAlertShown = false;

  // Water Tank Alerts
  if (waterPercent >= 90 && !fullAlertShown) {
    for (int i = 0; i < 3; i++) {
```

```
      digitalWrite(buzzerPin, HIGH);
      delay(300);
      digitalWrite(buzzerPin, LOW);
      delay(300);
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Tank is Full");
    lcd.setCursor(0, 1);
    lcd.print("Motor Turned Off");
    fullAlertShown = true;
    digitalWrite(relayPin, HIGH);
    delay(3000);
  }
  else if (waterPercent == 0 && !emptyAlertShown) {
    digitalWrite(buzzerPin, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Tank is EMPTY");
    delay(2000);
    digitalWrite(buzzerPin, LOW);
    emptyAlertShown = true;
  }
  else {
    // Show Water Level
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Water:");
```

```
lcd.print(waterPercent);
lcd.print("%");

lcd.setCursor(0, 1);
if (waterPercent <= 20) {
  lcd.print("Pump ON ");
} else {
  lcd.print("Pump OFF");
}
delay(2000);

// Show Temperature with symbol and text
lcd.clear();
lcd.setCursor(0, 0);
lcd.write(byte(0)); // Temperature symbol
lcd.print(" Temp:");
if (isnan(temperature)) {
  lcd.print("Err");
} else {
  lcd.print(temperature, 1); // 1 decimal place
  lcd.print((char)223);
  lcd.print("C");
}

// Show Humidity with symbol and text
lcd.setCursor(0, 1);
lcd.write(byte(1)); // Humidity symbol
lcd.print(" Hum:");
```

```
    if (isnan(humidity)) {
      lcd.print("Err");
    } else {
      lcd.print(humidity, 1); // 1 decimal place
      lcd.print("%");
    }
    delay(2000);
  }

  // Pump control
  if (waterPercent <= 20) {
    digitalWrite(relayPin, LOW);
  }
  else if (waterPercent >= 90) {
    digitalWrite(relayPin, HIGH);
  }

  delay(500);
}
```