



Faculty of Computer Science and Information Technology

**SMART WATER QUALITY MONITORING SYSTEM
BASED ON IOT**

NUR ALEESHA IZZATI BINTI JAFRI

Bachelor Degree of Software Engineering with Honours 2025

SMART WATER QUALITY MONITORING BASED ON IOT

NUR ALEESHA IZZATI BINTI JAFRI

This project is submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

SMART WATER QUALITY MONITORING BASED ON IOT

NUR ALEESHA IZZATI BINTI JAFRI

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Komputer Perisian dengan
Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I hereby declare that this project is my original work. I have not copied from others student's work or from any other sources except where due reference or acknowledgement is not made explicitly in the text, nor has any part had been written for me by another person.

.....

(NUR ALEESHA IZZATI BINTI JAFRI)

7th JULY 2025

1 Table of Contents

| | | |
|-------|--|----|
| 1 | Chapter 1: Introduction..... | 1 |
| 1.1 | Introduction..... | 1 |
| 1.2 | Problem Statements | 2 |
| 1.3 | Objectives | 3 |
| 1.4 | Methodology..... | 3 |
| 1.5 | Scope..... | 5 |
| 1.6 | Significance of Project..... | 5 |
| 1.7 | Project Outline | 6 |
| 1.8 | Project Schedule | 8 |
| 1.9 | Project Outcome | 9 |
| 1.10 | Summary..... | 9 |
| 2 | Chapter 2: Literature Review..... | 10 |
| 2.1 | Introduction..... | 10 |
| 2.2 | Background Study..... | 10 |
| 2.3 | Review on Existing Systems..... | 12 |
| 2.3.1 | Application of GPRS Technology in Water Quality Monitoring System..... | 12 |
| 2.3.2 | Water Quality Monitoring System Based on WSN Wu et al.,2010) | 15 |
| 2.3.3 | A Study on Traditional Water Quality Assessment Methods | 18 |
| 2.4 | Comparison between Proposed System..... | 19 |
| 2.5 | Comparison between Existing System and Proposed System | 21 |
| 2.6 | Reviews on Tools and Techonology Used | 22 |

| | | |
|-------|---|----|
| 2.7 | Summary..... | 27 |
| 3 | Chapter 3: Requirement Analysis and Design..... | 28 |
| 3.1 | Introduction..... | 28 |
| 3.2 | Project Methodology..... | 28 |
| 3.3 | Software Requirements..... | 31 |
| 3.4 | Hardware Requirements | 32 |
| 3.5 | Use Case Diagram | 33 |
| 3.5.1 | Use Case Description | 34 |
| 3.6 | Sequence Diagram | 37 |
| 3.7 | Block Diagram..... | 38 |
| 3.8 | Conclusion | 38 |
| 4 | Chapter 4: System Implementation..... | 39 |
| 4.1 | Overview..... | 39 |
| 4.2 | Hardware Implementation and Installation..... | 39 |
| 4.3 | Software Implementation..... | 41 |
| 4.3.1 | Arduino IDE | 41 |
| 4.3.2 | Xampp Implementation | 45 |
| 4.3.3 | Web System Dashboard | 48 |
| 4.3.4 | Data Handling | 49 |
| 4.4 | Prototype..... | 50 |
| 4.5 | System Modules..... | 50 |
| 4.5.1 | Database Module | 50 |
| 4.5.2 | Web Based Module | 51 |
| 4.5.3 | Microcontroller Integration Module | 51 |
| 4.5.4 | Sensor Data Acquisition Module | 51 |

| | | |
|-------|---|----|
| 4.6 | Snippet Code Implementation | 51 |
| 4.6.1 | Arduino Sensor Reading | 51 |
| 4.6.2 | ESP8266 for data transmission | 52 |
| 4.6.3 | PHP scripts | 52 |
| 4.7 | Summary | 53 |
| 5 | Chapter 5: Software Testing | 54 |
| 5.1 | Overview | 54 |
| 5.2 | System Testing | 54 |
| 5.3 | Functional Testing | 54 |
| 5.4 | Non-Functional Testing | 56 |
| 5.5 | Unit Testing | 59 |
| 5.6 | Integration Test Cases | 60 |
| 5.7 | Summary | 60 |
| 6 | Chapter 6: Conclusion and Future Works | 61 |
| 6.1 | Overview | 61 |
| 6.2 | Achievements | 61 |
| 6.3 | Limitations and Problem Encountered | 62 |
| 6.4 | Future Work | 63 |
| 6.5 | Summary | 63 |
| | References | 64 |
| | Appendix A | 66 |
| | Appendix B-Usability Text Questionnaire | 67 |

List of Tables

| | |
|---|----|
| TABLE 2.1 COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED | 21 |
| TABLE 3.1 SOFTWARE REQUIREMENTS | 31 |
| TABLE 3.2 HARDWARE REQUIREMENTS | 32 |
| TABLE 3.3 USE CASE DESCRIPTIONS FOR SENSOR READING | 34 |
| TABLE 3.4 USE CASE DESCRIPTION FOR DATA COLLECTION | 35 |
| TABLE 3.5 USE CASE DESCRIPTION FOR DATA ANALYSIS | 35 |
| TABLE 3.6 USE CASE DESCRIPTION FOR VIEW DATA COLLECTION | 36 |
| TABLE 5.1 FUNCTIONAL TESTING..... | 54 |
| TABLE 5.2 USABILITY TEST | 57 |
| TABLE 5.3 TEST CASES | 59 |
| TABLE 6.1 OBJECTIVES AND ACHIEVEMENTS..... | 61 |

Lists of Figures

| | |
|--|----|
| FIGURE 1.1 AGILE METHODOLOGY | 3 |
| FIGURE 2.1 GENERAL STRUCUTE OF THE SYSTEM. | 14 |
| FIGURE 2.2 GENERAL STRUCTURE OF THE SYSTEM. | 16 |
| FIGURE 2.3 SYSTEM DESIGN: SOFTWARE FLOW CHART OF GATEWAY NODE. | 17 |
| FIGURE 2.4 ARDUINO | 22 |
| FIGURE 2.5 TEMPERATURE SENSOR | 23 |
| FIGURE 2.6 LCD | 23 |
| FIGURE 2.7 TURBIDITY SENSOR | 24 |
| FIGURE 2.8 TOTAL DISSOLVED SOLIDS SENSOR | 24 |
| FIGURE 2.9 ESP8266 WI-FI MODULE | 25 |
| FIGURE 3.1 USE CASE DIAGRAM | 33 |
| FIGURE 3.2 SEQUENCE DIAGRAM | 37 |
| FIGURE 3.3 BLOCK DIAGRAM | 38 |
| FIGURE 4.1 SENSOR CONNECT TO ARDUINO | 40 |
| FIGURE 4.2 FULL CIRCUIT | 40 |
| FIGURE 4.3 DOWNLOAD ARDUINO IDE | 42 |
| FIGURE 4.4 ARDUINO CONNECT TO POWER VIA CABLE TYPE-B | 42 |
| FIGURE 4.5 CHOOSE BOARD | 43 |
| FIGURE 4.6 CHOOSE PORT | 44 |
| FIGURE 4.7 CODE TO SHOW THE SENSOR READING TO LCD | 44 |
| FIGURE 4.8 CODE TO SHOW THE SENSOR READING TO LCD | 45 |
| FIGURE 4.9 XAMPP INSTALLATION | 45 |
| FIGURE 4.10 XAMPP CONTROL PANEL | 46 |
| FIGURE 4.11 PHPMYADMIN | 47 |
| FIGURE 4.12 DATABASE AND TABLE | 47 |
| FIGURE 4.13 WEB SYSTEM DASHBOARD | 48 |
| FIGURE 4.14 NOTIFICATION FOR ALARM TRIGGER | 49 |
| FIGURE 4.15 CSV FILE FOR DATA EXPORTED | 49 |
| FIGURE 4.16 LCD EXAMPLE READING FOR TDS METER AND TURBIDITY SENSOR | 50 |
| FIGURE 5.1 AVERAGE USABLITY SCORES FOR EACH QUESTION | 58 |

List of Abbreviations

| | |
|------|------------------------------|
| IoT | Internet of Things |
| TDS | Total Dissolved Solids |
| NTU | Nephelometric Turbidity Unit |
| ESP | Espressif Systems Protocol |
| HTTP | Hypertext Transfer Protocol |
| LCD | Liquid Crystal Display |
| ADC | Analog to Digital Converter |
| DB | Database |
| UI | User Interface |

Abstract

Monitoring water quality is essential to maintaining human health and environmental safety. Traditional monitoring methods that depend on laboratory testing and manual water sampling are frequently expensive, time-consuming, and insensitive to changes in real time. This project proposes the creation of an Internet of Things (IoT)-based smart water quality monitoring system in order to get around these limitations. The system uses analog sensors that are connected to an Arduino UNO microprocessor to measure important water parameters like turbidity, temperature, and total dissolved solids (TDS). An ESP8266 Wi-Fi module sends the data gathered from these sensors to a web-based dashboard for real-time visualisation. The system was built in several stages, including hardware configuration, sensor integration, data transfer, and web interface creation, due to the Agile development technique. To assess the overall user experience, data transfer dependability, and sensor reading accuracy, functional and usability testing were carried out. The outcomes showed that the system operates dependably for temperature, turbidity, and TDS measurements, with precise and prompt data updates shown on the web dashboard and local LCD.

Pemantauan kualiti air amat penting untuk memastikan kesihatan manusia dan keselamatan alam sekitar. Kaedah pemantauan tradisional yang bergantung kepada ujian makmal dan pensampelan air secara manual sering kali melibatkan kos yang tinggi, memakan masa, dan tidak responsif terhadap perubahan secara masa nyata. Projek ini mencadangkan pembangunan sistem pemantauan kualiti air pintar berasaskan Internet of Things (IoT) bagi mengatasi kekangan ini. Sistem ini menggunakan sensor analog yang disambungkan kepada mikropemproses Arduino UNO untuk mengukur parameter penting kualiti air seperti kekeruhan (turbidity), suhu, dan jumlah pepejal terlarut (TDS). Modul Wi-Fi ESP8266 menghantar data yang dikumpul daripada sensor-sensor ini ke papan pemuka berasaskan web untuk visualisasi secara masa nyata. Sistem ini dibangunkan secara berperingkat merangkumi konfigurasi perkakasan, integrasi sensor, pemindahan data, dan pembangunan antara muka web, berdasarkan kaedah pembangunan Agile. Ujian fungsi dan kebolegunaan dijalankan bagi menilai ketepatan bacaan sensor, kebolehpercayaan pemindahan data, dan pengalaman pengguna secara keseluruhan. Hasil ujian menunjukkan bahawa sistem ini berfungsi dengan baik dalam pengukuran suhu, kekeruhan, dan TDS, dengan kemas kini data yang tepat dan pantas dipaparkan pada papan pemuka web dan LCD setempat.

1 Chapter 1: Introduction

1.1 Introduction

Monitoring the quality of the water is crucial to ensuring safety and purity of water resources like lakes and rivers. Conventional methods of testing water quality can be time-consuming and hard work requiring both physical sampling and laboratory analysis. In order to address this, we can utilise an Internet of Things (IoT)-based system for monitoring water quality. This system employs sensors to collect data in real-time on critical water parameters such as temperature, turbidity and total dissolved solids.

In Malaysia, the water problem has become a national issue, particularly in the state of Selangor during the past ten years. 10.7 million people, or 31.4% of Malaysia's total population, live in Selangor, which includes the Federal Territories of Kuala Lumpur and Putrajaya (Department of Statistics Malaysia, 20243Q). With an estimated 5.3% increase in gross domestic production per capita, the state's total GDP is RM490.60 billion. Nearly 90% of the residents of Selangor, the most populated and developed state in Malaysia, reside around the three main river basins of the state, which are the Selangor, Klang, and Langat rivers.

The Selangor Water Management Authority (SWMA) reported that 13 water treatment plants (WTPs) were shut down in 2020 as a result of raw water quality violations, making the three river basins flooded areas over the past ten years. Up to 1.14 million people were impacted by water outages in 10 of the 13 incidents (Air Selangor, 2020). Therefore, it is necessary to control the water quality in these river basins effectively.

Monitoring the quality of river water on a regular basis is essential for preventing and controlling pollution. It gives information and data to enable and allow prompt action in identifying possible pollution sources and identifying areas where poor quality is occurring. In order to minimise disturbance to the water supply, an emergency plan is put in place once the contaminants cross the required limit(Loi et al., 2022).The moitoring of water quality then acts as a starting point for the management of water resources in an environmentally friendly way(Loi et al., 2022). Regular inconsistencies in the quality of river water can be shown by the monitoring data (Loi et al., 2022

1.2 Problem Statements

Water pollution and contamination pose significant risks to ecosystems, human health, and industries that rely on clean water. According to the World Health Organisation (WHO), over 485,000 diarrheal deaths occur annually due to drinking water contaminated with microorganisms (Zhang et al., 2023). Ensuring clean water quality is critical for public safety and sustainable water management. However, existing water quality monitoring methods face several challenges:

A. Inefficiency of Traditional Water Monitoring Methods

Conventional water quality monitoring relies heavily on manual sampling and laboratory analysis, which is time-consuming, costly, and prone to delays (Zulkifli et al., 2022). The lack of real-time monitoring results in slow detection of contamination, making it difficult to take immediate corrective actions.

B. Limited Data Analysis and Accessibility

Many traditional water quality monitoring systems focus solely on data collection without proper analysis or real-time visualization (Zhang et al., 2021). The inability to remotely access data limits decision-making capabilities, particularly for organizations managing large water bodies such as lakes and reservoirs.

C. Challenges in Data Reliability and Transmission

Current monitoring systems often face data loss and inaccuracies due to transmission issues and power outages (Danh et al., 2020). Without a reliable and accessible monitoring system, water quality assessments may be incomplete, leading to potential risks in water safety and resource management.

1.3 Objectives

The primary objectives of this project are as follows:

- i. To design an efficient Internet of Things (IoT)-based water quality monitoring system that incorporates sensors to measure important water parameters (temperature, turbidity and total dissolved solids)
- ii. To develop a web-based system that enables real-time monitoring to water quality information.
- iii. To evaluate the effectiveness and reliability of the water quality monitoring system

1.4 Methodology

It is crucial to choose a proper methodology for every project, it can determine the success or failure of the project. The most suitable methodology being used in this project is Agile methodology. Agile methodology is an adaptable and iterative approach to software development and project management that places a high value on client satisfaction, teamwork, and flexibility. Agile, as compared to traditional linear approaches, uses iterative cycles called sprints to enable continuous improvement and incremental advancement. This method works best for projects that are dynamic and need to be flexible enough to adjust to changes or new information as they are being developed. This guarantees that the finished system successfully satisfies user needs and permits early testing and quick modifications to problems. Figure 1.1 shows the process of Agile Methodology.



Figure 1.1 Agile Methodology

The **Planning phase** focuses on identifying project objectives, requirements, and deliverables. The project is divided into smaller, more manageable tasks that will be completed in sprints or iterations. To prevent future issues, this phase makes sure that all required parts, including software and hardware, are designed effectively. Potential risks, including difficulties with setting up sensors or network connectivity, are also taken into account during this phase, and mitigation techniques are created.

During the **Design phase** the system architecture is developed, including IoT sensors, a microcontroller, and a web-based dashboard for real-time data visualization. To determine how users will engage with the system, web interface wireframes are made. Establishing a flexible, flexible structure that allows simple development in the event that more features are needed later on is the aim of this phase.

The **Development phase** is where iterative cycles (sprints) produce incremental improvements in the system. Here is the sprints:

- a. Sprint 1: Setup and configure IoT sensors to collect water quality data. Each sensor is tested individually to ensure accuracy and proper integration with the microcontroller.
- b. Sprint 2: Develop the web-based interface for real-time data monitoring. The interface includes a dashboard for displaying sensor data.
- c. Sprint 3: Integrate cloud storage for historical data analysis and alerts. Data collected from sensors is stored inside the Mysql database securely to allow long-term tracking.
- d. Sprint 4: Test system reliability and optimize performance. This involves debugging issues related to sensor readings, network connectivity, and user interface responsiveness.

The **Testing phase** makes sure the system works the way it should. While integration testing makes sure that every system component functions as a whole, unit testing verifies each component's performance separately. To evaluate the system's accuracy and dependability, nonfunctional and functional testing will be done.

During the the system was tested in a controlled environment using regular water to simulate real conditions. After all hardware components were assembled and the web-based dashboard was fully functional, the sensors were placed in water to verify if real-time data could be captured, transmitted, and displayed correctly. This stage confirmed the end-to-end functionality of the system, from physical sensing to online visualization.

1.5 Scope

The scope of this project is to create an Internet of Things (IoT)-based water quality monitoring system that gives a thorough awareness of water quality by continuously monitoring vital water parameters like temperature, turbidity, and dissolved solids. The system includes IoT sensors, a microcontroller, and a web-based platform for real-time data visualization. This system is designed to assist individuals or organizations managing water bodies, such as lakes or reservoirs.

1.6 Significance of Project

In order to overcome the shortcomings of conventional water quality monitoring techniques, this project is extremely valuable. Conventional monitoring techniques frequently depend on manual sampling, which takes a lot of time, gives results slowly, and might not capture changes in water quality in real time. Due to these restrictions, reactions to contamination may be delayed, which could endanger human health or harm the environment. Real-time monitoring of water quality metrics is provided by the suggested IoT-based system, facilitating quicker and better decision-making.

This project's ability to completely automate the water quality monitoring process, which eliminates the need for manual labour and provides continuous, real-time data, is one of its main advantages. Critical parameters like temperature, turbidity, and total dissolved oxygen are regularly measured thanks to the integration of IoT sensors, enabling the prompt diagnosis of any abnormalities. By addressing the problem of slow detection in conventional approaches, this feature makes sure that users can act quickly to stop the spread of water contamination.

The web-based system offers data visualisation interface that makes it simple for users to view sensor readings and examine historical data via XAMPP database. Decision-makers, environmental organisations, and water management groups can react quickly to any contamination or unusual readings when water quality is remotely monitored, preventing additional damage and guaranteeing water safety. Additionally, by safely storing gathered data in cloud storage, this solution improves data reliability. This increases the precision of long-term water quality tests by removing the possibility of data loss from transmission mistakes or power outages.

To sum up, this project has many advantages, such as automated monitoring, real-time data availability, accessibility through a web system, and data saving in XAMPP databases.

1.7 Project Outline

Chapter 1 will focus on Introduction. An introduction describes a particular about the introduction of the project, problem statement, scope, objectives, brief methodology, significances of project, project outline and summary.

Then, Chapter 2 is literature review. A literature review describes the research topic about water quality monitoring technique and other technologies that is being used. This chapter will also describe the software and technology used for this project implementation.

Next, Chapter 3 discusses the methodology adopted for this project. The Agile methodology was selected to guide the development process, allowing for iterative planning, continuous feedback, and adaptive progress through multiple development cycles. This chapter explains the Agile workflow phases, including planning, design, implementation, testing, and evaluation. It also outlines the tools, technologies, and hardware components used, particularly focusing on the procedures involved in experimenting with Arduino for water quality monitoring.

Chapter 4 focuses on the System Implementation, which includes both the hardware and software aspects of the project. The hardware section describes how the sensors which are TDS, turbidity, and temperature were connected and configured with the microcontroller. The software section explains how the Arduino code was developed to capture sensor data and

transmit it to a web server. It also covers the PHP backend, MySQL database, and the web-based dashboard interface. This chapter includes code snippets and explanations to demonstrate how the system components work together to achieve real-time water quality monitoring.

Chapter 5 discuss about SoftwareTesting, covering functional testing, non-functional testing, and unit testing. This chapter evaluates whether the system meets its intended requirements and performs reliably under various conditions. Test cases are documented to show how each function of the system was validated, while non-functional aspects such as performance and usability are also discussed.

Finally, Chapter 6 concludes the report by summarizing the project outcomes and highlighting the lessons learned. It also discusses future enhancements, such as the development of a mobile app version, improved sensor calibration, and the inclusion of more advanced data analytics. This This chapter reflects on the overall development experience, acknowledge limitations and provides recommendations for further improvement.

1.8 Project Schedule

Tue 1/10/24 Add tasks with dates to the timeline

| | Ta | Task Name | Duration | Start | Finish |
|----|----|--|-----------|--------------|--------------|
| 1 | ✚ | Final Year Project | 220 days? | Tue 1/10/24 | Thu 31/7/25 |
| 2 | ✚ | First Semester: Proposal Preparation Phase | 75 days? | Tue 1/10/24 | Sun 12/1/25 |
| 3 | ✚ | Identify and consult with Supervisor | 9 days? | Tue 1/10/24 | Fri 11/10/24 |
| 4 | ✚ | Initial draft of the brief proposal | | | |
| 5 | ✚ | Prepare and finalize the brief proposal based on supervisor feedback | 7 days? | Sat 12/10/24 | Sat 19/10/24 |
| 6 | ✓ | Brief Proposal Submission for approval from examiners | 1 day | Sat 12/10/24 | Sat 12/10/24 |
| 7 | ✚ | Submission of the Approved Brief Proposal | 1 day | Sun 20/10/24 | Sun 20/10/24 |
| 8 | ✚ | Revise the proposal based on examiners feedback | 6 days? | Mon 21/10/24 | Sun 27/10/24 |
| 9 | ✚ | Do correction for the brief proposal (If any) | | | |
| 10 | ✚ | Start drafting and do the Full Proposal | | | |
| 11 | ✚ | Receive feedback and comments from examiners | 1 day | Mon 28/10/24 | Mon 28/10/24 |
| 12 | ✚ | Finalize the Full Proposal | 0 days? | Sun 13/10/24 | Sun 13/10/24 |
| 13 | ✚ | Submission of the Full Proposal | 1 day | Thu 14/11/24 | Thu 14/11/24 |
| 14 | ✚ | Prepare and Submit Chapter 1 (Introduction) | 5 days? | Thu 14/11/24 | Wed 20/11/24 |
| 15 | ✚ | Submission of the Chapter 1 | 1 day | Thu 21/11/24 | Thu 21/11/24 |
| 27 | ✚ | Research and Write Chapter 2 (Literature Review) | 16 days? | Thu 21/11/24 | Thu 12/12/24 |
| 28 | ✚ | Submission of the Chapter 2 | 1 day | Fri 13/12/24 | Fri 13/12/24 |
| 29 | ✚ | Research and Write Chapter 3 (Methodology) | 17 days? | Fri 13/12/24 | Sat 4/1/25 |
| 30 | ✚ | Submission of the Chapter 3 | 1 day | Sun 5/1/25 | Sun 5/1/25 |
| 31 | ✚ | Compile and Finalize FYP 1 Report | 7 days | Sun 5/1/25 | Sat 11/1/25 |
| 32 | ✚ | Prepare all of the Final Report and Paper Submission | | | |

| | Ta | Task Name | Duration | Start | Finish |
|----|----|--|----------|--------------|--------------|
| 27 | ✚ | Research and Write Chapter 2 (Literature Review) | 16 days? | Thu 21/11/24 | Thu 12/12/24 |
| 28 | ✚ | Submission of the Chapter 2 | 1 day | Fri 13/12/24 | Fri 13/12/24 |
| 29 | ✚ | Research and Write Chapter 3 (Methodology) | 17 days? | Fri 13/12/24 | Sat 4/1/25 |
| 30 | ✚ | Submission of the Chapter 3 | 1 day | Sun 5/1/25 | Sun 5/1/25 |
| 31 | ✚ | Compile and Finalize FYP 1 Report | 7 days | Sun 5/1/25 | Sat 11/1/25 |
| 32 | ✚ | Prepare all of the Final Report and Paper Submission | | | |
| 33 | ✚ | Submit FYP 1 Final Report for the Assessment | 1 day | Sun 12/1/25 | Sun 12/1/25 |
| 34 | ✚ | | | | |
| 35 | ✚ | Second Semester: Implementing and Testing Phase | 33 days? | Tue 1/4/25 | Thu 15/5/25 |
| 36 | ✚ | Begin Development and Implementation of the IoT-based water quality monitoring system and Apps | 30 days | Tue 1/4/25 | Sat 10/5/25 |
| 37 | ✚ | Finding and Collecting all of the hardware and software needs | | | |
| 38 | ✚ | Testing the IoT System and mobile application for data accuracy and functionality | 5 days | Sun 11/5/25 | Thu 15/5/25 |
| 39 | ✚ | Test all of the hardware functionality | | | |
| 40 | ✚ | Test the application features and functionality | | | |
| 41 | ✚ | System testing report | 1 day | Thu 15/5/25 | Thu 15/5/25 |
| 42 | ✚ | Deployment of the System and Presentation Preparations | 2 days | Tue 1/7/25 | Wed 2/7/25 |
| 43 | ✚ | Deployed system and final presentation slides | | | |
| 44 | ✚ | Final Presentation of the Final Year Project | 1 day? | Tue 1/7/25 | Tue 1/7/25 |
| 73 | ✚ | | | | |

1.9 Project Outcome

The main result of this project is the development of an Smart Water Quality Monitoring System Based on IoT that continuously measures important water quality indicators such as turbidity, temperature, and dissolved oxygen. This system addresses the shortcomings of conventional water quality monitoring techniques, which often rely on manual sampling and delayed laboratory analysis, making them inefficient and prone to errors. A web-based dashboard provides real-time monitoring capabilities.

XAMPP integration guarantees data accessibility and dependability. By providing real-time data, the IoT solution enables users to monitor water quality more effectively, allowing for rapid intervention and informed decision-making in the event of any irregularities. Continuous monitoring ensures a more precise and dynamic evaluation of water conditions, contributing to better resource preservation and environmental management.

Additionally, the system includes features for visualizing real-time data visualization on the dashboard and the data is saved inside the xampp database. Overall, this project provides an efficient and accessible solution for water quality monitoring, enhancing environmental sustainability and public health through the use of IoT technology.

1.10 Summary

In conclusion, this chapter discusses the introduction of proposed system. The proposed system is about developing Smart Water Quality Monitoring System Based on IoT to use. This system had been proposed to overcome the limitations and issues with the current conventional approach. The background of this project is briefly introduced in this chapter. The introduction includes an introduction of project, problem statement, scope, objectives, methodologies, significances of project, project outline, project schedule and expected outcome of the project and lastly summary.

2 Chapter 2: Literature Review

2.1 Introduction

This chapter explores an in-depth review of the various techniques and technologies used for water quality monitoring, from conventional to innovative approaches. Conventional manual sampling techniques have long been the norm for identifying contaminants and examining characteristics like turbidity, temperature, and total dissolved oxygen. These techniques require collecting water samples from lakes, rivers, or reservoirs for laboratory study. Even though these methods provide precise and thorough results, they are less useful for ongoing monitoring since they are time-consuming, labour-intensive, frequently lack real-time data capabilities, and prone to human error. The aim is to critically analyze the strengths and weaknesses of these methods, identify trends in research, and uncover gaps that the proposed system seeks to address. This chapter will also describe the software and technology used for this project implementation.

2.2 Background Study

Monitoring water quality is essential to sustainable development, public health, and environmental management. A variety of ecosystems, agriculture, industry, and human use all depend on clean water. However, because of urbanisation, industrialisation, and agricultural drainage, which introduce dangerous materials like pesticides, heavy metals, and microbiological contaminants into water bodies, water pollution has grown in importance. Effective monitoring methods are crucial for prompt intervention and management of these pollutants because they represent serious threats to aquatic life, ecosystems, and human health.

Manual sampling techniques, which involve gathering water samples from multiple areas and bringing them to labs for examinations, have historically been used to monitor the quality of water. Although these techniques produce precise findings for parameters, they are inevitably labour-intensive, time-consuming, and have a narrow scope (Zulkifli et al., 2022). For instance, because there isn't constant monitoring, they frequently miss abrupt changes in the quality of the water. Additionally, human errors during data collection, management, and transportation could affect the dependability of manual sampling.

By bringing real-time, automated systems that combine sensors, microcontrollers, and cloud-based platforms, the Internet of Things (IoT) has completely changed the way that water quality is monitored (AlMetwally et al., 2020). Key water quality metrics including temperature, turbidity, and dissolved oxygen can be continuously monitored thanks to IoT-based devices, which can instantly notify users when limits are crossed. By automating data collecting, decreasing human error, and enabling stakeholders to access real-time data remotely, these systems overcome the drawbacks of conventional approaches (Glassgow et al., 2004). IoT systems are also cost-effective for large-scale deployments because they can be expanded to monitor several sites at once.

IoT-based monitoring systems still face obstacles in spite of these developments, such as the requirement for dependable internet connectivity, initial setup expenses, and data security issues. Additionally, not all IoT solutions are made to be affordable, which limits their usability for people in impoverished nations or for small-scale applications. These difficulties show how effective systems that strike a compromise between precision, cost, and usability are required.

In conclusion, efficient IoT-based systems that can continuously check water quality in real time have replaced manual, labour-intensive approaches. However, there are still limitations in terms of cost-effectiveness and accessibility, especially for small-scale or decentralised users. By creating an intelligent IoT-based water quality monitoring system that blends real-time accuracy with cost and ease of use, our project seeks to close these gaps.

2.3 Review on Existing Systems

2.3.1 Application of GPRS Technology in Water Quality Monitoring System

The current approach to water quality monitoring used by XueYe reservoir involves sampling on-site and laboratory analysis. This study provides an autonomous water quality monitoring system based on GPRS after analysing the main issues with water quality monitoring. The system's general architecture and network architecture, as well as the data monitoring center's GPRS access method, are all based on GPRS technology and take into account the characteristics of water quality monitoring.

The data processing and monitoring centre receives the water quality parameters gathered by the multi-parameter water quality test via a mobile device's GPRS wireless connection network. Due to the system's automatic transmission, and processing of water quality indicators, production efficiency and economic benefits are significantly increased. Research has demonstrated that GPRS technology can function effectively in the intricate world of unmonitored low water quality. More especially, it can be applied to the collection point, where data transmission automatically creates the field of water analysis equipment data transmission and monitoring.

In order to continuously and in real-time monitor water bodies, especially those in the Xueye Reservoir, this research investigates an advanced water quality monitoring system that makes use of GPRS (General Packet Radio Service) technology. Important water quality indicators, such as turbidity, dissolved oxygen, and others, can be measured by the system's multi-parameter sensors. These sensors provide quick analysis and prompt reactions to changes in water quality by transferring data to a centralised monitoring centre via wireless GPRS modules.

This GPRS-based system's main goal is to overcome the disadvantages of conventional water quality monitoring, namely labour-intensive procedures and delayed findings. It provides real-time sequential monitoring, which guarantees timely detection of pollution or environmental disasters and improves the effectiveness of water quality tests. The solution guarantees scalability for large-scale deployments and drastically lowers operating costs by using public mobile networks for data transmission.

Additionally, the system includes data visualisation capabilities that let stakeholders track trends in water quality and spot patterns across time. Making well-informed decisions for catastrophe prevention and environmental management is aided by this predictive skill. It is a

workable option for reservoirs, rivers, and other water bodies that are at risk of pollution because of its design, which guarantees adaptation to a variety of environmental situations.

The GPRS network, the data processing and monitoring centre, and the water quality data collection device make up the three main parts of the automatic GPRS-based water quality monitoring system.

Sensors on the water quality data collection device collect data on important water quality indicators in real time. Additionally, it gets orders from the monitoring centre and follows them to carry out tasks. Data transmission to the monitoring centre through the mobile communication network is made easier by this instrument's connection to the GPRS terminal. The sent data is received at the monitoring centre, processed by a data server, and then stored in a database. Users can now obtain historical or current water quality data thanks to this. At operator stations connected via a local area network, the data can be displayed in a variety of formats. Additionally, field users can use PCs or mobile devices with web interfaces or wireless GPRS connections to access the monitoring center's server.

The wireless network, the central monitoring station, and the field-based sensors are all seamlessly integrated by this system architecture, providing users with scalable and adaptable access to water quality data for efficient monitoring and decision-making. Figure 2.1 shows the general structure of the system.

In conclusion, by combining wireless connection, real-time data collecting, and predictive analytics, the GPRS-based water quality monitoring system marks a substantial improvement over conventional technique. This system serves as an example of how contemporary technology may improve water management by providing a practical, scalable, and affordable means of maintaining water quality and protecting ecosystems.

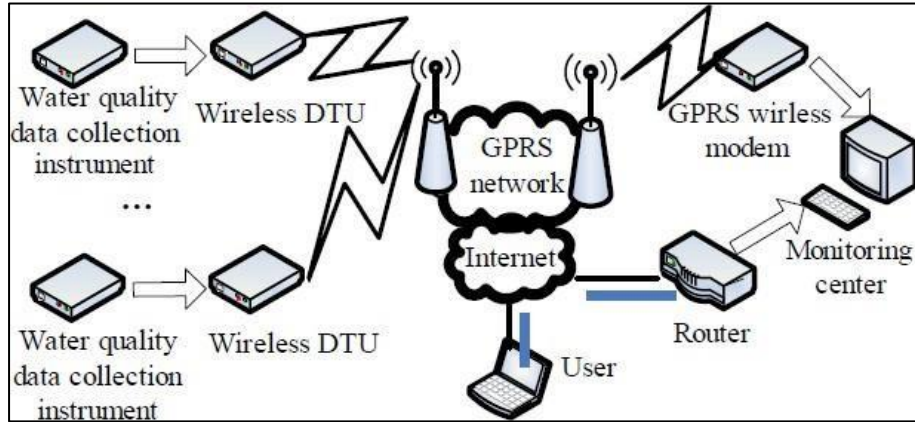


Figure 2.1 General Strucute of the System.

(Wu et al.,2010)

2.3.2 Water Quality Monitoring System Based on WSN Wu et al.,2010)

The creation of an Internet of Things (IoT)-based water quality monitoring system that makes use of Wireless Sensor Networks (WSN) for ongoing, real-time water body monitoring is the main goal of this study. The system offers a low-cost, scalable, and energy-efficient way to monitor water quality parameters like turbidity, temperature, and dissolved oxygen by integrating Zigbee technology for communication with GPRS (General Packet Radio Service) for data transmission. By automating the procedure and providing real-time data gathering and analysis, the system overcomes the drawbacks of conventional water quality monitoring techniques, which are frequently time-consuming and labour-intensive.

Zigbee technology, a wireless communication protocol developed for low-power, low-data-rate applications, is the basis of the system. Zigbee is perfect for large-scale water quality monitoring installations since it uses the IEEE 802.15.4 standard and can enable scalable networks with up to 64,000 nodes. The mesh design of the network ensures ongoing monitoring even over long distances or in difficult situations by enabling data to hop between nodes, increasing the system's dependability and range. Coordinator nodes in the system aggregate data from router and terminal nodes to oversee the network. Multi-parameter water quality sensors installed on the terminal nodes gather information on a range of water quality metrics.

Coordinator nodes, router nodes, and terminal nodes are the three main component types included in the hardware architecture. The coordinator node acts as the network's focal point, using GPRS connection to send and receive data between the remote data centre and the wireless sensor network. In order to ensure that information may be sent across long distances, router nodes relay data between terminal nodes and the coordinator. Water quality sensor data must be gathered by terminal nodes, which must then use the CC2430 microcontroller's Analog-to-Digital Converter (ADC) to transform the analogue sensor data into digital information and send it to the coordinator. Using an off-chip Real-Time Clock (RTC) to control sampling intervals, the terminal nodes run in low-power modes and use very little current which typically less than 110 μ A.

The software system is designed around the Zigbee protocol stack, which includes the physical, MAC, network, security, and application layers. Data collection, transmission to the data centre, and node status updates are all managed by the coordinator node software. By establishing a communication link between the coordinator and the terminal nodes, the router

node software guarantees smooth data transfer. In order to provide effective and dependable connection with the rest of the system, the sensor node software is in charge of collecting, processing, and delivering water quality data on a regular basis in accordance with the Zigbee protocol.

On the backend, the information is sent to a distant monitoring facility for processing, archiving, and analysis. Real-time data visualisation and historical trend analysis are made possible by the software tools included in the system. By giving stakeholders access to the most recent data on water quality, these solutions help avert possible problems with water contamination. This system offers an economical, scalable solution for ongoing water quality monitoring in a variety of contexts by fusing low-power wireless sensors networks, effective data transmission technologies, and real-time monitoring capabilities.

To sum up, the suggested system combines state-of-the-art technologies like WSN, Zigbee, and GPRS to provide an automated, economical, and energy-efficient solution for real-time water quality monitoring. It overcomes the disadvantages of conventional techniques, guaranteeing accurate, real-time data collecting and giving interested parties the resources they need to make prompt, well-informed decisions about the management of water resource.

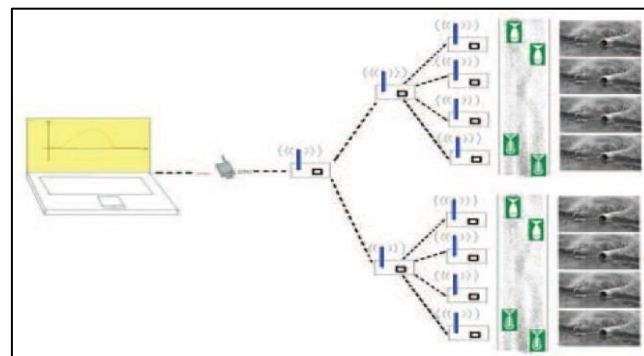


Figure 2.2 General Structure of the System.

(He & Zhang., 2012)

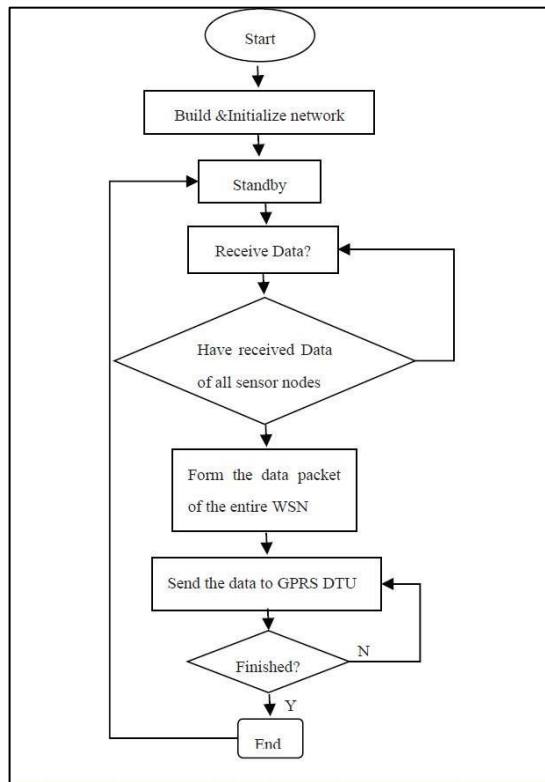


Figure 2.3 System Design: Software Flow Chart of Gateway node.

(He & Zhang., 2012)

2.3.3 A Study on Traditional Water Quality Assessment Methods

This study investigates the effectiveness of conventional techniques for evaluating water quality in controlling pollution brought on by both natural occurrences like erosion and floods as well as human activities like industrial discharge and agricultural runoff. In order to evaluate different physical, chemical, and microbiological aspects of water, these techniques mostly rely on laboratory analysis and on-site water sample collection. Although traditional methods are thought to be accurate for examining particular water quality measures, the study assesses how well they work to solve contemporary water management issues and points out their drawbacks.

Conventional monitoring techniques are useful for scientific research and regulatory compliance since they are excellent at giving high precision for particular parameters like turbidity, and dissolved oxygen. They enable thorough testing, frequently employing tried- and-true techniques including electrometric analysis, gravimetric measurements, and titration. Because these techniques are internationally standardised, the outcomes are generally recognised by regulatory agencies and industries. But in spite of their accuracy and dependability, they have serious limitations that make them unsuitable for large-scale or real-time water monitoring.

The study pointed up a number of shortcomings in conventional approaches. Their time-consuming nature is one of the main obstacles; it frequently takes days for sample collection, transportation, and laboratory analysis to yield useful data. The effects of water pollution may worsen as a result of this delay until remedial action is taken. These techniques are also labour-intensive, needing trained workers for laboratory and fieldwork, which raises operating expenses. Additionally, traditional methods are unable to deliver real-time data, which restricts their capacity to efficiently track dynamic changes in water quality. The reliability of the results might also be impacted by problems like sampling mistakes, which include improper handling during collection or transit.

In order to analyse the efficacy of conventional approaches, the study looks at a number of popular evaluation methodologies. The Single-Factor Assessment Method, for example, ignores the combined impacts of several contaminants while concentrating on the most important one. Although the Nemerow Pollution Index takes into account averages for all pollutants, it places too much emphasis on extreme numbers, which could skew the results. In

a similar vein, the Comprehensive Pollution Index assesses a large number of factors while making the frequently impractical assumption that each one is equally important. Advanced techniques like Principle Component Analysis (PCA) and Fuzzy Comprehensive Evaluation are also included in the study. These techniques simplify complicated datasets and offer more lucid insights, although they may not work well with nonlinear or severe data circumstances.

The study comes to the conclusion that although old methods are still useful for accurate, one-time assessments, they are less appropriate for meeting the demands of contemporary water quality monitoring due to their inherent drawbacks, which include long reaction times, high labour costs, and a lack of scalability. In order to provide more effective water management in the face of expanding environmental issues, the research highlights the growing need for creative, real-time solutions that integrate automation, cost-effectiveness, and scalability to supplement or replace conventional approaches.

2.4 Comparison between Proposed System

The proposed system uses Internet of Things (IoT) technologies to transform the monitoring of water quality. By offering automatic, continuous, and real-time data collecting and analysis, it overcomes the drawbacks of conventional techniques. The system is a robust and user-friendly solution designed for effective and scalable water quality monitoring. It combines sophisticated hardware elements including sensors, a microcontroller, and a communication module with a web-based platform for real-time data visualization and analysis.

Numerous sensors that measure important aspects of water quality are at the core of the system. These include a temperature sensor to track temperature changes, a total dissolved oxygen sensor to measure dissolved salt or ion levels for salinity studies and a turbidity sensor to measure the clarity of water by detecting suspended particles. To give a precise picture of the water's status, these sensors continuously gather data in real time. An Arduino UNO microcontroller, which serves as the central control unit, then processes the data. The Arduino regulates the whole flow of data from the sensors, guarantees the accuracy of readings, and gets them ready for transmission.

The system makes use of the ESP8266 Wi-Fi module for wireless connectivity, sending the processed data to a cloud-based platform. A online dashboard provides remote access to

real-time water quality data, enabling trend analysis and ongoing monitoring. When the water quality exceeds predetermined levels, the web interface offers reading for each water monitoring data and alarm notifications. Professionals on-site may also rapidly check sensor data thanks to the integrated LCD display, guaranteeing prompt intervention if needed.

In contrast to conventional techniques that depend on laborious and prone to delays manual sampling and laboratory analysis, our system offers real-time insights that facilitate prompt decision-making. Data gathering automation minimises errors related to manual monitoring and drastically lowers the cost of human labour. Additionally, by utilising a centralised web platform, several users—such as researchers, environmental organisations, and businesses can obtain data on water quality from various areas without requiring the retrieval of physical data.

The proposed system has a number of benefits over conventional techniques. By removing the delays brought on by manual sampling and laboratory testing, it guarantees real-time monitoring. Because the system is completely automated, labour expenses and human error are decreased. The system's scalability makes it possible to deploy several sensor nodes throughout vast bodies of water, which makes it appropriate for a range of uses, including industrial water monitoring, aquaculture, and reservoir management. A web-based dashboard also improves accessibility to view the data directly and the data is saved inside XAMPP database. The system offers an affordable option for ongoing water quality monitoring since it reduces long-term operating expenses related to labour and equipment maintenance.

In conclusion, this Internet of Things (IoT)-based water quality monitoring system offers a thorough, dependable, and effective approach to water quality management. Its incorporation of cutting-edge technology guarantees precision, scalability, and accessibility, satisfying the requirements of applications in public health, agriculture, and environmental management.

2.5 Comparison between Existing System and Proposed System

Table 2.1 Comparison between Existing System and Proposed

| System | Proposed System | Application of GPRS Technology in Water Quality Monitoring System | Water Quality Monitoring System Based on WSN | A Study on Traditional Water Quality Assessment Methods |
|---|-----------------|---|--|---|
| Provide real time update | ✓ | ✓ | ✓ | × |
| Provide trend analysis | × | ✓ | ✓ | × |
| Display the data on computer | ✓ | ✓ | ✓ | × |
| Store the data inside database/ History | ✓ | ✓ | × | ✓ |
| Web Based Dashboard | ✓ | × | × | × |

2.6 Reviews on Tools and Technology Used

In order to provide precise data collection, effective transmission, and insightful analysis, the IoT-based water quality monitoring system's installation necessitates the integration of multiple hardware and software components. The software guarantees seamless connection and intuitive data visualisation, while the hardware serves as the system's structural core.

1) Hardware

1) Arduino UNO



Figure 2.4 Arduino

The Arduino UNO serves as the primary microcontroller for the system. It reads data from the water quality sensors, processes this information, and transmits it to the communication module for cloud integration. Its simplicity, availability of libraries, and compatibility with various sensors make it an ideal choice for prototyping and small-scale applications.

2) Temperature Sensor



Figure 2.5 Temperature Sensor

This sensor measures the water's temperature, a critical parameter for assessing thermal conditions in aquatic environments. The data is essential for applications such as aquaculture, as temperature impacts aquatic life and chemical reactions in water

3) LCD



Figure 2.6 LCD

The LCD acts as a display interface for immediate local monitoring. It shows real-time data readings from sensors, making the system more user-friendly and accessible for field technicians or on-site personnel.

4) Turbidity Sensor



Figure 2.7 Turbidity Sensor

The turbidity sensor evaluates water clarity by detecting suspended particles. This data helps identify contamination from sediments, industrial discharge, or other pollutants, making it vital for monitoring drinking water quality and environmental health.

5) Total dissolved oxygen Sensor



Figure 2.8 Total dissolved Solids Sensor

The total dissolved solids sensor measures dissolved salt and ion levels. This parameter is critical in evaluating water salinity, particularly for agriculture, aquaculture, and freshwater sources.

6) ESP8266 Wi-Fi Module



Figure 2.9 ESP8266 Wi-Fi Module

The ESP8266 enables wireless data transmission by connecting the Arduino UNO to the internet. It ensures seamless data transfer from the microcontroller to a cloud platform like ThinkSpeak for real-time monitoring and visualization

2) Software

a. Web Based Dashboard

- HTML, CSS, JavaScript – Standard web technologies for structuring, styling, and scripting.
- -CSS/Bootstrap -Frameworks to style the dashboard.
- -C++/C- The Arduino UNO is programmed using C++/C, which allows for controlling hardware components like sensors and the ESP8266 module. The code ensures proper data acquisition, processing, and communication. For instance, it sets up reading intervals for sensors, calibrates measurements, and manages data transmission.
- Arduino IDE- Used to write and upload programs to the Arduino UNO and ESP8266 modules. It also provides a serial monitor for real-time debugging and testing.
- MySQL/XAMPP- A MySQL database is used to store sensor data, allowing users

to retrieve historical records and analyze trends over time. This ensures efficient data management and long- term storage for water quality monitoring.

- JavaScript (Node.js & Express.js)- Node.js and Express.js handle the backend operations, ensuring efficient data transmission between the ESP8266 module and the web dashboard. These technologies allow users to fetch real-time and historical water quality data securely.

2.7 Summary

In order to understand the difficulties and developments in water quality monitoring, case studies and current systems are essential components of this study. These illustrations provide insightful information about the advantages, constraints, and real-world application of existing technologies. The literature offers a thorough grasp of the problems pertaining to water quality management by examining systems like the GPRS-based water quality monitoring system, IoT-based systems that use WSN (Wireless Sensor Networks), and conventional manual monitoring techniques.

The GPRS-based system research, for example, demonstrates how technology can enhance water quality monitoring by facilitating real-time data collection, prompt replies, and economical scalability via wireless communication. Traditional approaches, on the other hand, emphasise the limitations of manual sampling and the related delays and errors that impede effective decision-making, notwithstanding their high accuracy. These real-world examples, in turn, highlight the necessity of cutting-edge technologies like Internet of Things (IoT)-based systems that provide automated alarms, remote data access, and continuous monitoring, thereby solving many of the drawbacks of the conventional methods.

The literature offers real-world examples of successful implementations by integrating existing systems, providing not just theoretical insights but also scalable, practical solutions that function in many settings. The suggested solution for water quality monitoring can be improved thanks to these case studies, which also enable us to spot gaps in existing technologies, especially those pertaining to the price and usability of IoT systems. With their potential to improve public health, agriculture, and environmental sustainability, these examples greatly deepen our understanding of how IoT systems might improve water quality management across a range of industries and applications

3 Chapter 3: Requirement Analysis and Design

3.1 Introduction

This chapter describes the approach taken to create the suggested Internet of Things-based water quality monitoring system. It demonstrates a consistent and logical understanding of the system's needs, the procedures involved, and the implementation process flow. The system is made to efficiently accomplish the goals through agile methodology. In order to demonstrate the system's features and operations, this chapter also contains detailed illustrations and descriptions, including use cases, sequence diagrams, and activity diagrams.

3.2 Project Methodology

This project is a good fit for the Agile methodology's basic principles of collaborative work, iterative development, and adaptability. It offers a structured yet adaptable framework for creating and implementing a working prototype that tackles the issue of irregular water quality monitoring. Agile makes sure that progress is quantifiable and meaningful throughout the project lifecycle by concentrating on delivering a certain set of features in each sprint. This method was selected because combining IoT hardware and software is experimental and may involve unexpected issues that need to be resolved quickly, such as sensor incompatibilities, connectivity problems, and hardware constraints. The project include design, development, testing, deployment, and planning. Every stage was carefully organised according to Agile principles, guaranteeing that every sprint produced a working increment of the system as a whole.

During the Planning stage, the project's main goals were established, one of which was to create and deploy an Internet of Things (IoT)-based water quality monitoring system that could provide real-time sensor data on a specially designed via the web dashboard. By examining the planned features, performance standards, and environmental factors, the system requirements were identified. By identifying the main elements involved, such as the kinds of sensors such as turbidity, TDS, and temperature, the Arduino microcontroller, and the software technologies needed for the frontend and backend, the project's scope is outlined. The tasks were divided into small stages and dispersed among four major development sprints. Because each sprint concentrated on a different aspect of the system, the development process was able to stay structured and focused on objectives.

During the **Design Phase**, A design of the hardware and software components will be created throughout the design phase. The hardware design included the necessary resistors and modules such as I2C for the LCD display, the power supply configuration, and the actual placement and cabling of the IoT sensors to the Arduino board. To guarantee constant performance, special attention was given to the selection of dependable components. The system architecture for the software component was created to enable HTTP requests to be sent back and forth between the Arduino microcontroller and a web server.

During the design stage, a number of important features were included to improve the overall system's usability and performance. The automatic data handling mechanism was one of the most important additions. Effective database size management was required since the system continuously gathers sensor readings. A PHP script that is scheduled to run once a week and exports all data older than seven days into a CSV file was used to construct a backup system. To guarantee optimal speed and storage, the old data is removed from the database and this backup file is kept in a specific backup folder in htdocs.

To increase user interaction with the dashboard, a number of the front-end enhancements were also will be created in addition to data handling. To alert people when the temperature rises above a certain safety threshold such as 40°C and above, a temperature alarm system will be implemented. In order for users to control the alarm sound according to their preferences, a toggle switch will be added. Additionally, a button for restarting is located above the turbidity card. With the use of this feature, users can manually remove all of the current sensor readings and then quickly reload the dashboard with the most recent data. Adding "three previous readings" for each sensor which are temperature, turbidity, and TDS was an additional feature that makes it simple for users to keep an eye on short term patterns without having to visit the complete history page.

The Development phase is where iterative cycles (sprints) produce incremental improvements in the system. Here are the sprints:

- **Sprint 1:** Put sensors and Arduino hardware together, program the microcontroller to record temperature, total dissolved solids and turbidity and verify the unprocessed sensors' outputs. Special focus is placed on ensuring that sensor readings align with known standards and that data from the Arduino can be easily processed for future visualization.

- **Sprint 2:** Attach an LCD screen so that local water quality parameters can be seen in real-time. During this stage, sensor data is also clarified by converting raw values into user-readable formats, improving the utility of the local display. Field tests may also be conducted to validate readability under real-world conditions.
- **Sprint 3:** Development of web-based systems. HTML, CSS, JavaScript, Node.js, and Express.js are used in the development of the custom dashboard. To provide real-time updates, the ESP8266 Wi-Fi module is set up to send sensor data to a MySQL database. To guarantee dependable data connection, connectivity issues including irregular network transmission are found and fixed.
- **Sprint 4:** This sprint focused on optimizing the system's stability and responsiveness. The ESP8266 and Arduino code was refined to ensure efficient WiFi reconnections, reduce noise in sensor readings, and improve timing between data transmissions. The frontend UI was also polished for clarity and ease of use.

The **testing phase** makes sure that every system component operates as intended and satisfies performance requirements. Each module was thoroughly tested to make sure it operated as intended. Each parts, including the sensors, the Arduino code, and every PHP script, will undergo unit testing. This will helped in the early identification and isolation of defects during the development cycle. Testing will be done both functionally and non-functionally. Functional testing confirmed that important functions including data storage, accurate sensor readings, and real-time updates operated as planned. Non-functional testing evaluated the performance, usability, and dependability of the system under various water conditions and with repeated use.

The Deployment Phase involves installing a fully functional prototype of the system at a designated site or using water sample from lakes for real-time monitoring. During the deployment stage, the system will be put together completely and tested in. To imitate real world use cases, all sensors will be installed and immersed in water samples. The system was run over WiFi in a local network setting, and the Arduino was fuelled separately by a USB power source. By using a local IP address to access the dashboard, the system will be verified if the system could gather, process, and display real-time data on water quality.

3.3 Software Requirements

Table 3.1 below shows the software requirements for the proposed system. The software that we are going to have in the proposed system is summarized below.

Table 3.1 Software Requirements

| Software | Requirement |
|---------------------------|---|
| C++ | Arduino is built on top of C++, a popular and widely utilised programming language. The code is developed in C++ and includes additional functions and special methods. |
| Arduino IDE | The Arduino Software (IDE), also known as the Arduino Integrated Development Environment, includes a text editor for writing code, a message box, a text console, a toolbar with buttons for frequently used tasks, and a number of menus. In order to upload programs and interact with the Arduino hardware, it connects to it. |
| Node.js & Express.js | Node.js is used to build the backend of the web-based system, while Express.js serves as the web framework for handling API requests. This combination ensures real-time data communication between the microcontroller and the web dashboard. |
| MySQL Database | MySQL is used to store sensor readings and historical data. This database allows for efficient querying, ensuring users can retrieve past water quality records and analyze trends over time. |
| HTML, CSS, and JavaScript | These frontend technologies are used to develop the web-based dashboard. HTML structures the content, CSS styles the interface, and JavaScript provides interactivity for real-time data visualization. |
| XAMPP | A local server environment used for testing MySQL database functionality before deployment. |

3.4 Hardware Requirements

Table 3.2 below shows the hardware requirements for the proposed system. The hardware that we are going to have in the proposed system is summarized below.

Table 3.2 Hardware Requirements

| Hardware | Description |
|-------------------------------|---|
| Arduino UNO | It reads data from the water quality sensors, processes this information, and transmits it to the communication module for cloud integration. |
| Temperature Sensor | This sensor measures the water's temperature, a critical parameter for assessing thermal conditions in aquatic environments. |
| LCD | The LCD acts as a display interface for immediate local monitoring. It shows real-time data readings from sensors. |
| Turbidity Sensor | The turbidity sensor evaluates water clarity by detecting suspended particles to identify contamination. |
| Total dissolved Solids Sensor | The total dissolved solids sensor measures the water's electrical. Total dissolved solids to assess dissolved salt and ion levels. |
| ESP8266 Wi-fi Module | The ESP8266 enables wireless data transmission by connecting the Arduino UNO to the internet. |

3.5 Use Case Diagram

The water quality monitoring system is shown in the use case diagram in Figure 3.1. Furthermore, the scientists will be the actors in the suggested system. The lower-level functions, which are the water quality sensors employed in the system, are among the functions, along with data collecting, display data collection, data analysis, and sensors reading.

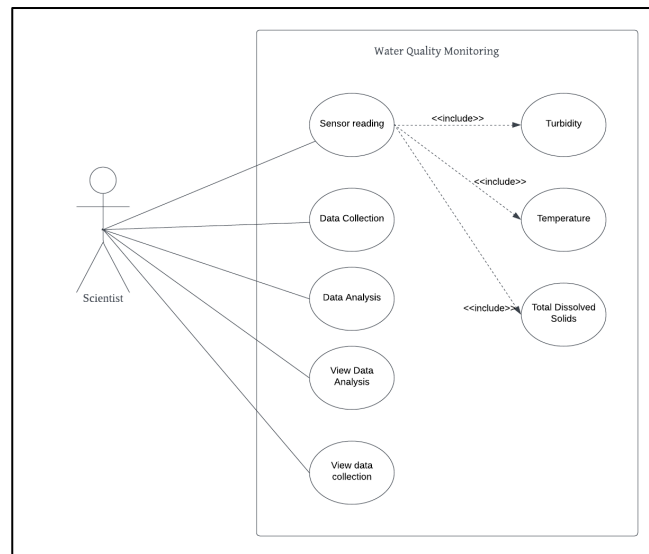


Figure 3.1 Use Case Diagram

3.5.1 Use Case Description

The use case descriptions details is shown in shown in Table 3.3 – Table 3.6.

Table 3.3 Use Case Descriptions for Sensor Reading

| | |
|--------------------------|---|
| No. | UC1 |
| Use Case Name | Sensor Reading |
| Actor | Sensors |
| Description | The sensors collect raw data from the water, including turbidity, temperature, and total dissolved solids |
| Precondition | Sensors are connected to the Arduino and powered on. |
| Post condition | Data is read from sensors and sent to Arduino for processing. |
| Flow of the event | <ol style="list-style-type: none">1. Sensors are activated.2. Sensors gather data on turbidity, temperature, and total dissolved solids3. Data is passed to Arduino for processing. |

Table 3.4 Use Case Description for Data Collection

| | |
|--------------------------|--|
| No. | UC2 |
| Use Case Name | Data Collection |
| Actor | Arduino, Sensors |
| Description | Data gathered from the sensors is stored temporarily within the Arduino or transmitted directly to Web System. |
| Precondition | Sensors have sent raw water quality data to Arduino. |
| Post condition | Data is successfully stored, display on LCD and sent to Web System. |
| Flow of the event | <ol style="list-style-type: none"> 1. The Arduino receives raw data from the sensors. 2. Data is formatted and transmitted for storage or display on the LCD and sent to Web system dashboard. |

Table 3.5 Use Case Description for Data Analysis

| | |
|--------------------------|---|
| No. | UC3 |
| Use Case Name | Data Analysis |
| Actor | Arduino |
| Description | The system analyzes the collected data. |
| Precondition | Raw data is successfully collected and stored. |
| Post condition | Analyzed data is available for viewing by the scientist. |
| Flow of the event | <ol style="list-style-type: none"> 1. Arduino processes raw data. 2. Patterns or abnormalities in the data are identified. 3. Analysis results are made available. |

Table 3.6 Use Case Description for View Data Collection

| | |
|--------------------------|---|
| No. | UC5 |
| Use Case Name | View Data Collection |
| Actor | Scientist |
| Description | The scientist views raw, real-time water quality data either through the LCD and the Web System. |
| Precondition | Data has been collected from the sensors and sent to the display system. |
| Post condition | Raw data is displayed in real-time for evaluation. |
| Flow of the event | <ol style="list-style-type: none"> 1. Data from the sensors is displayed on the LCD or sent to Web System. 2. The user can access to Web System to view current water quality metrics in real-time. |

3.6 Sequence Diagram

The sequence diagram shows how five classes which are water, Arduino, sensors, LCD, and Dashboard interact with one another. The administrative system (Scientist) will use the sensors (within the water sample) to conduct a water quality check as shown in the sequence diagram of the complete pollution detection procedure. After receiving a sensor request, data is gathered from the sensors and sent to the Arduino. Following sensor collection, the Arduino UNO processes the data and shows it locally on the LCD screen so that field technicians or users on-site can view it right away. The ESP8266 Wi-Fi module simultaneously sends the processed data to the Web System Dashboard. Figure 3.2 shows the sequence diagram.

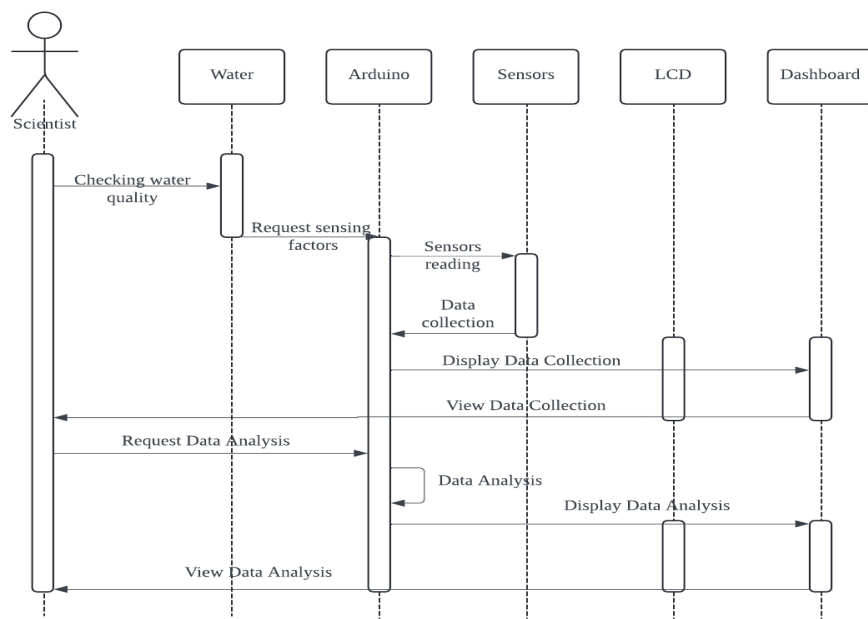


Figure 3.2 Sequence Diagram

3.7 Block Diagram

The system's central processing unit is the Arduino UNO. All linked sensors provide raw data, which it then transforms into a format that may be used. Before sending the data to the output devices, the Arduino verifies that the readings are accurate and trustworthy. It serves as a bridge between the digital data transmission components and the physical sensors.

The ESP8266 Wi-Fi Module transmits the processed data to the cloud for remote storage and accessibility, enabling Internet of Things capability. End users can view the reading on the dashboard web system and the data will be saved inside XAMPP.

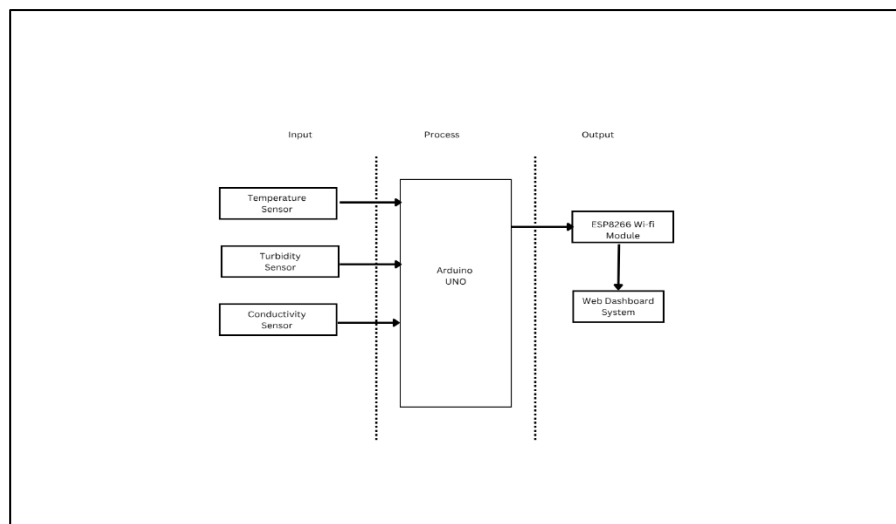


Figure 3.3 Block Diagram

3.8 Conclusion

This chapter provides a solid framework for applying the Agile methodology to the system's design and development. Agile's incremental and iterative methodology guarantees that development is adaptable, effective, and in line with customer requirements at every stage. Agile's frequent cycles of feedback and collaboration improve the system's flexibility and user pleasure. Furthermore, clear visualisations of processes and interactions are provided via diagrams such use case diagrams, sequence diagrams, and block diagrams, which foster an overall comprehension of the system and allows constant improvements throughout development.

4 Chapter 4: System Implementation

4.1 Overview

System implementation is the process of developing the proposed system into a working prototype or putting the proposed system into process. In order to create the proposed system, hardware and software must be acquired, developed, and implemented at this stage of development. The system undergoes testing after the development phase is over.

4.2 Hardware Implementation and Installation

To implement the IoT-based water quality monitoring system, several hardware components were required to support sensor connectivity, data transmission, and on-site data display. A computer with at least an Intel Dual-Core processor, 2GB RAM, and running Microsoft Windows 7 or later was used for programming and uploading code via the Arduino IDE.

The essential hardware components used in this project include:

- Arduino UNO Board
- ESP8266 Wi-Fi Module (NodeMCU)
- Liquid Crystal Display (LCD) 1602 with I2C Module
- TDS (Total Dissolved Solids) Sensor
- Turbidity Sensor SKU: SEN0189
- DS18B20 Waterproof Temperature Sensor

In this project, the Arduino UNO functions as the main controller that collects data from all water quality sensors. The sensors which are turbidity, TDS, and temperature are connected to the analog and digital input pins on the Arduino. An I2C-based LCD module is used to display real-time sensor readings, providing immediate feedback to users on-site.

As shown in Figure 4.1 shows all sensors were wired through the Arduino board. The readings were processed directly on the microcontroller. The ESP8266 Wi-Fi module was used to transmit the collected sensor data to a web-based system.

Power was supplied to the Arduino via USB from the computer, and the ESP8266 was powered through the 3.3V output of a voltage regulator or a dedicated 3.3V pin. Figure 4.2 shows the complete circuit setup, including sensor connections, LCD display, and data transmission to the web system via the ESP8266.

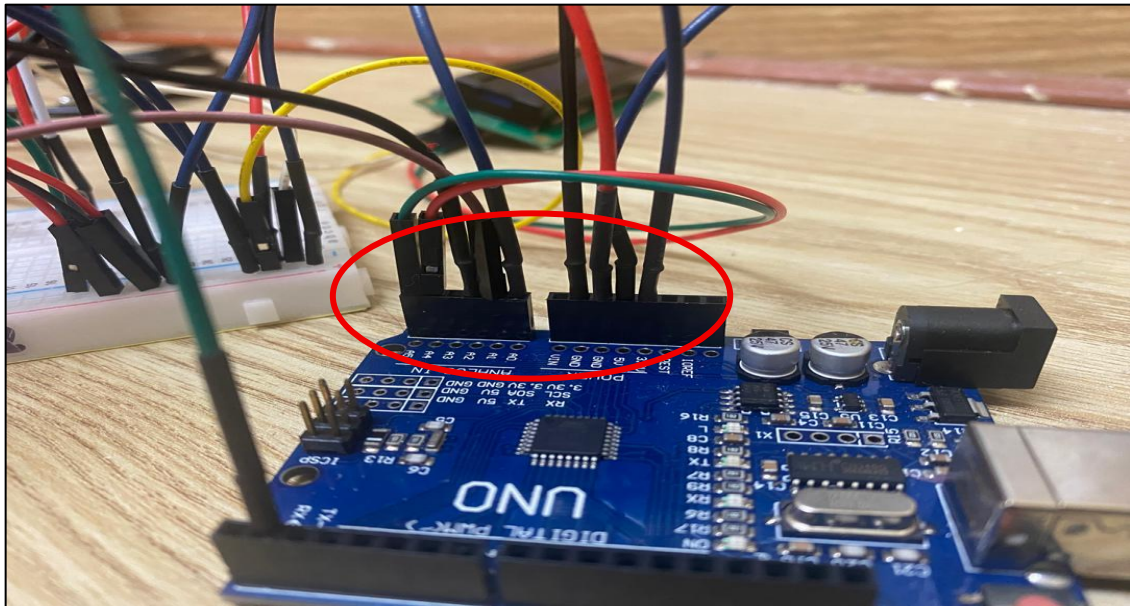


Figure 4.1 Sensor Connect to Arduino

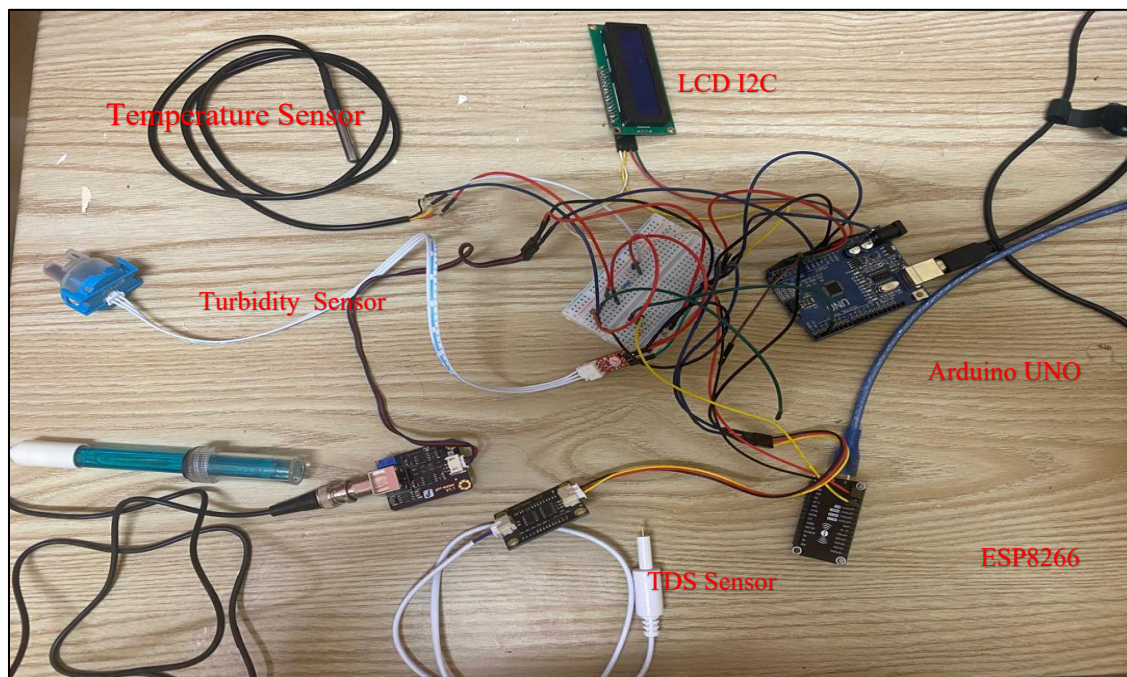


Figure 4.2 Full Circuit

For the hardware implementation, there are 3 steps of installation which are Circuit

Connections, Uploading Arduino Code and Powering the System.

A) Circuit Connections. Sensor to Arduino

1. Turbidity Sensor: VCC to 5V, GND to GND, Analog output to A1
2. TDS Sensor: VCC to 5V, GND to GND, Analog output to A2
3. Thermistor: One side to 5V, other side to A3 and 4.7k Ω resistor to GND
4. LCD I2C Display: SDA to A4, SCL to A5
5. ESP8266: RX to TX (Pin 1), TX to RX (Pin 0), GND to GND, VCC to 3.3V

B) Uploading Arduino Code

1. Disconnect the ESP8266 from TX/RX pins before uploading.
2. Open Arduino IDE, select the correct COM port.
3. Paste the Arduino sensor reading code. (Refer Source Code File)
4. Click Upload.
5. Reconnect ESP8266 after upload.

C) Uploading ESP8266 Code

1. Disconnect Arduino RX/TX if connected directly
2. Use a USB to Serial adapter or FTDI module to upload the ESP8266 code
3. Open Arduino IDE, select the board (e.g., NodeMCU 1.0) and COM port
4. Paste the ESP8266 code and upload. (Refer Source Code File)

4.3 Software Implementation

4.3.1 Arduino IDE

The Arduino IDE (Integrated Development Environment) is a free software for Arduino programming. It can be downloaded at <https://www.arduino.cc/en/software/>. Linux, Mac, and Windows are all capable of running it. The official Arduino IDE download website is shown in Figure 4.3.



Figure 4.3 Download Arduino IDE

The Arduino IDE needs to be set up correctly before any code is uploaded to the Arduino UNO. The Arduino UNO must first be connected to the computer via a USB Type-B connector as shown in Figure 4.4. The Arduino board's power LED will light up after connection, signalling that power is being received and the board is prepared for communication.

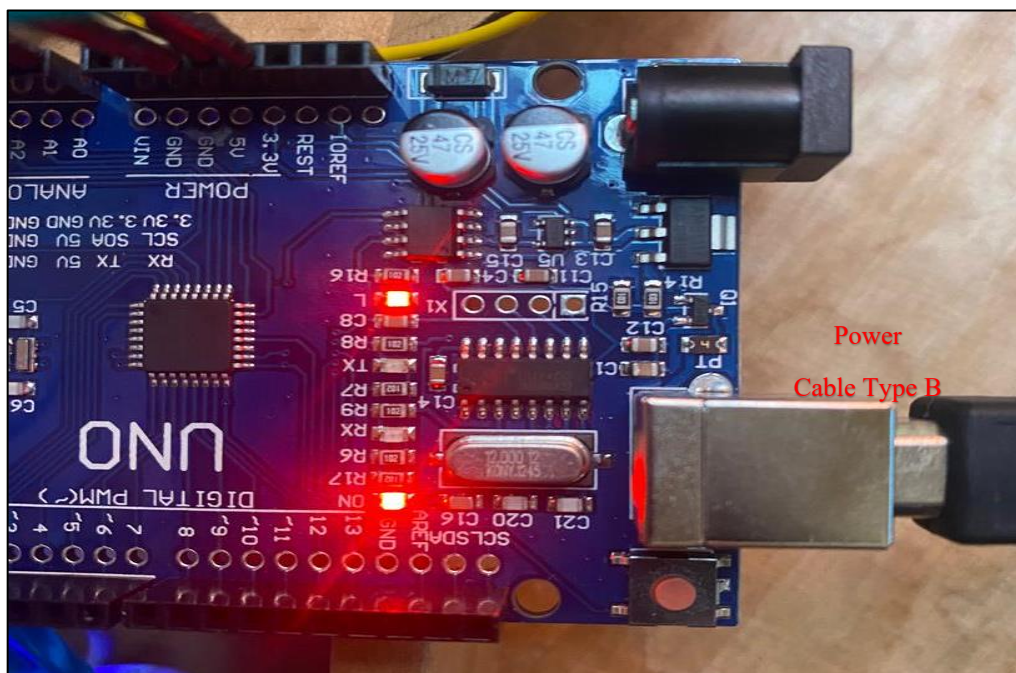


Figure 4.4 Arduino Connect to Power via cable type-B

To ensure the correct connection after starting the Arduino IDE, it's important to choose the appropriate board type. As shown in Figure 4.5, this can be completed by going to the "Tools" menu, selecting "Board," and then picking "Arduino UNO" from the list of available boards. Choosing the right board choice ensures that the IDE makes use of the device's compatible upload connection and compiler settings.

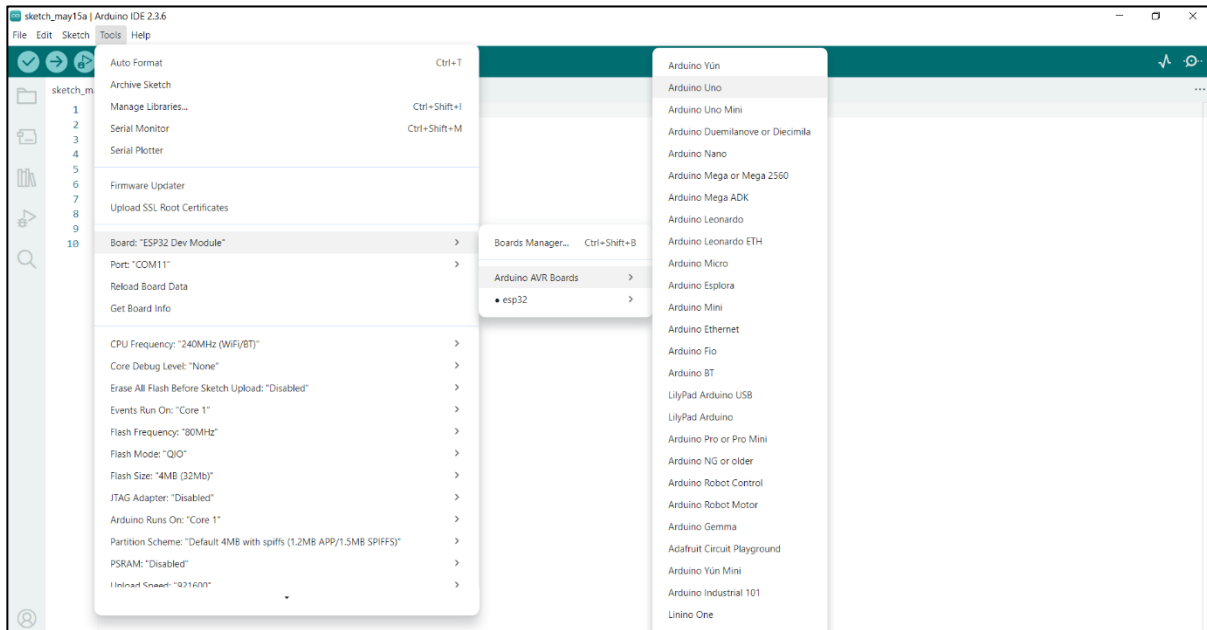


Figure 4.5 Choose Board

The next step is to choose the communication port. Click "Port" under the "Tools" menu, then select the COM port that matches the Arduino that is attached as shown in Figure 4.6. Usually, the name "Arduino UNO" or a similar designation like "COM3" or "COM4" is used to identify the correct port. To determine which port is the right one to use if there are several listed and it's uncertain, the user can unplug and reconnect the Arduino to see which port disappear and then return.

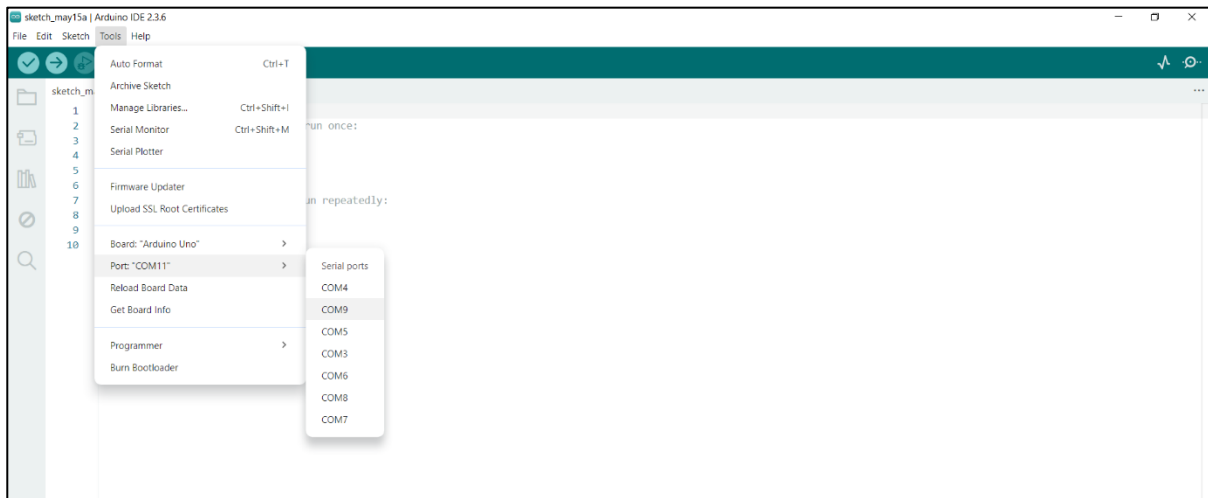


Figure 4.6 Choose Port

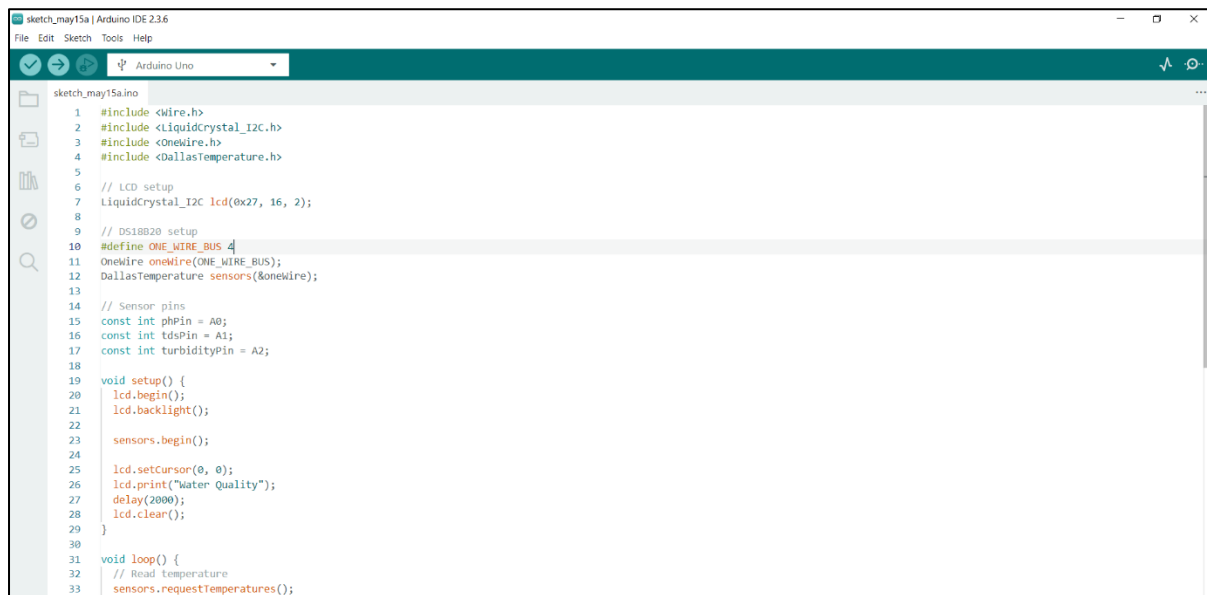


Figure 4.7 Code to show the sensor reading to lcd

```

sketch_may15a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
sketch_may15a.ino
30
31 void loop() {
32 // Read temperature
33 sensors.requestTemperatures();
34 float temperature = sensors.getTempCByIndex(0);
35
36 // Read other sensors (mock conversion - replace with real formulas if needed)
37 int phValue = analogRead(phPin);
38 float ph = (phValue * 5.0 / 1023.0) * 3.5; // example conversion
39
40 int tdsValue = analogRead(tdsPin);
41 int tds = map(tdsValue, 0, 1023, 0, 1000); // example TDS range in ppm
42
43 int turbValue = analogRead(turbidityPin);
44 int turbidity = map(turbValue, 0, 1023, 0, 3000); // example turbidity range in NTU
45
46 // Display first row
47 lcd.setCursor(0, 0);
48 lcd.print("pH:");
49 lcd.print(ph, 1);
50 lcd.print(" TDS:");
51 lcd.print(tds);
52
53 // Display second row
54 lcd.setCursor(0, 1);
55 lcd.print("Turb:");
56 lcd.print(turbidity);
57 lcd.print(" T:");
58 lcd.print(temperature, 1);
59
60 delay(2000);
61 }
62

```

Figure 4.8 Code to show the sensor reading to lcd

Figure 4.7 and 4.8 shows the example code and it will be upload to Arduino Uno board to test the sensor reading on the LCD. If there is now readings shows on the LCD, check the wiring and the code again.

4.3.2 Xampp Implementation

XAMPP installation can be download from Apache Friends website <https://www.apachefriends.org/download.html> as shown in Figure 4.9.

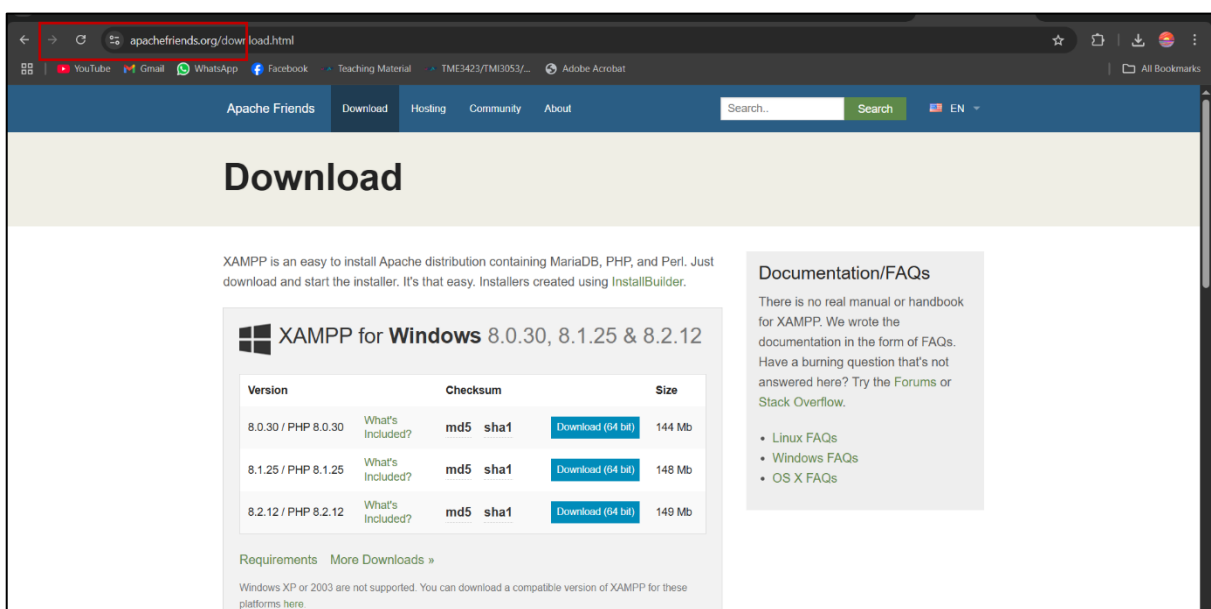


Figure 4.9 Xampp Installation

Similar with most Windows application installers, launch the XAMPP installer after downloading it from the website. All of the components that come with the XAMPP package can be controlled using the XAMPP control panel, as seen in Figure 4.10. All of the XAMPP modules can be operated from the control panel by simply clicking the start or stop button on each module.

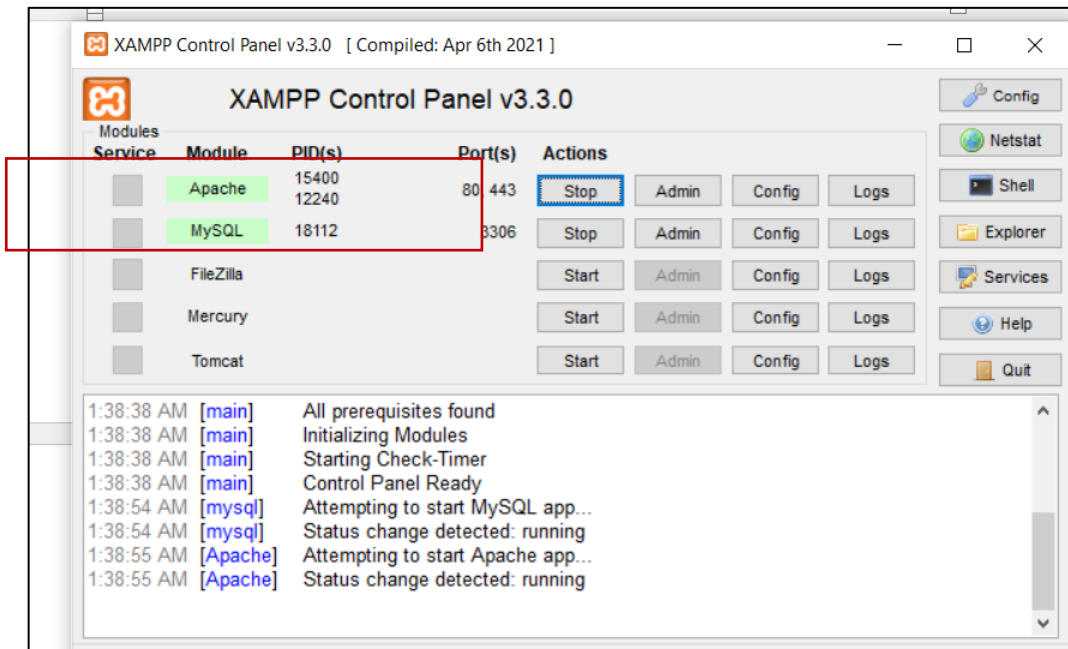


Figure 4.10 XAMPP Control Panel

Once the Apache and MySQL is run, we will access the database by using web browser and typing localhost/phpmyadmin/ as shown in Figure 4.11.

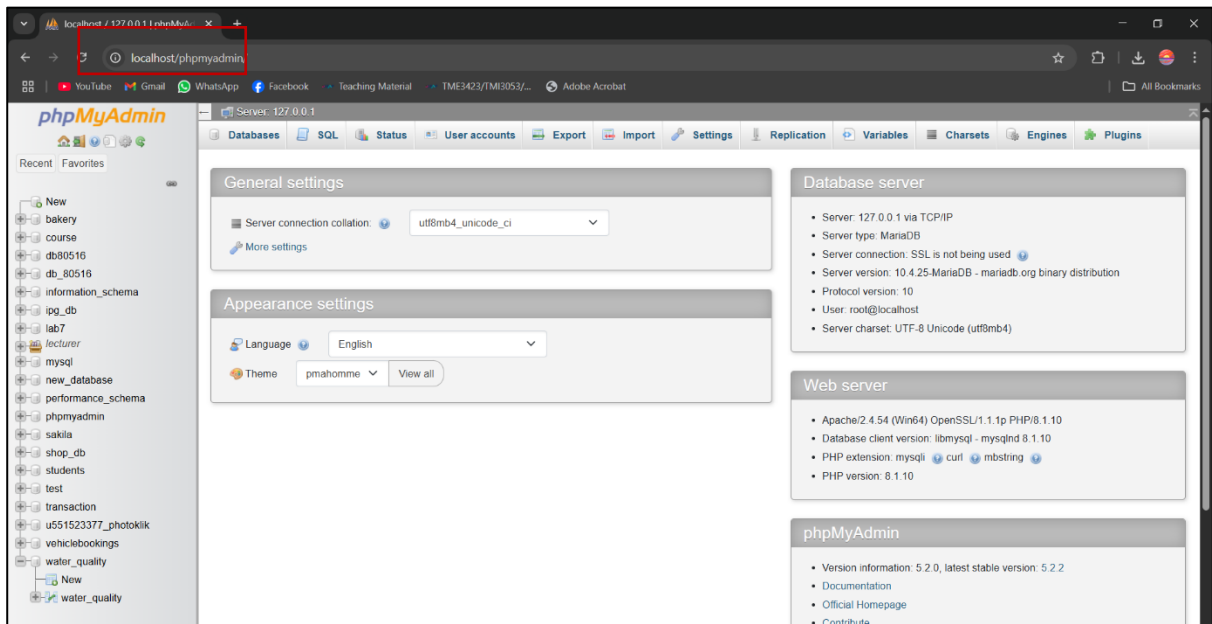


Figure 4.11 phpMyAdmin

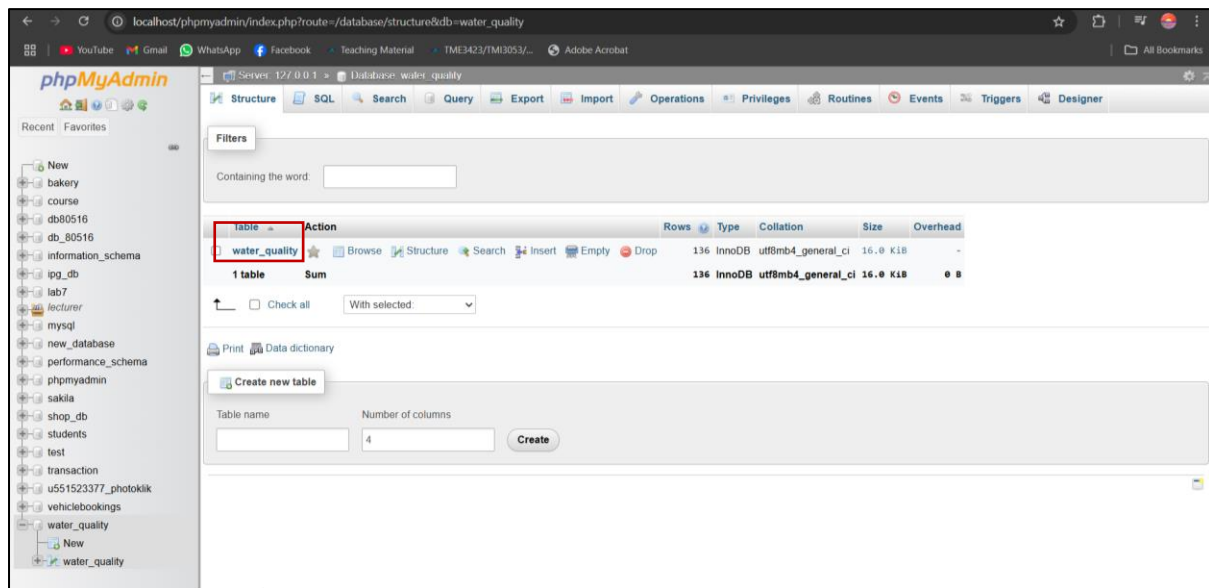


Figure 4.12 Database and Table

In order to store the sensor readings, a new database must first be created. To achieve this, select the "New" option from the left sidebar, type a suitable name, such as water_quality, and press the "Create" button. The real sensor values must be stored in a new table after the database has been built. The TDS level, water temperature, turbidity level, timestamp for each reading, and a unique ID for each row were all stored in a table called water_quality, which had six fields created for this project.

4.3.3 Web System Dashboard

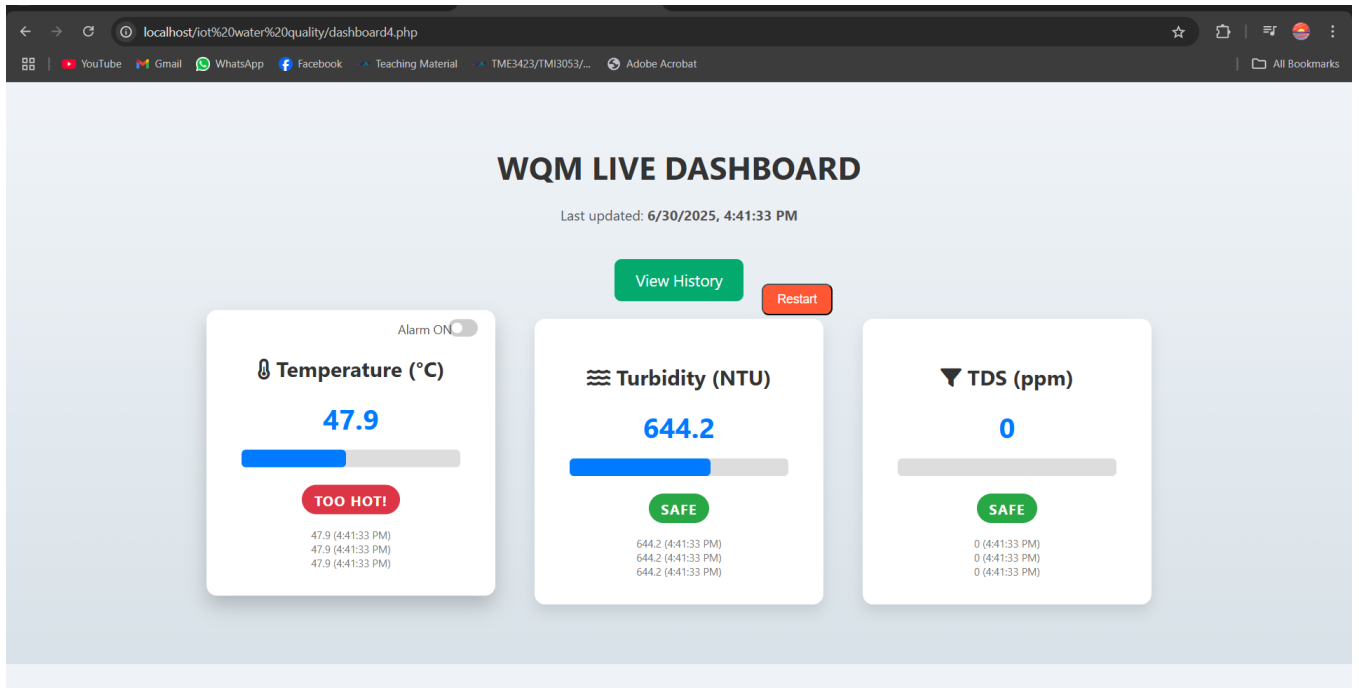


Figure 4.13 Web System Dashboard

Figure 4.13 shows the example of the web-based dashboard to show the readings for the sensor. A number of interactive features were added to the dashboard. When temperature readings surpass the danger level, the temperature alarm system sounds a warning and displays visual indicators. The temperature card has a toggle switch that users can use to adjust this function. The restart button is another feature that makes it simpler for users to easily reset the dashboard in the event of inaccurate or delayed data by clearing all existing values and refreshing them after a brief delay. Additionally, the last three recorded values and their timestamps are shown on each sensor card. From the main interface, users may now see real time changes or shifts in water quality parameters through this feature.

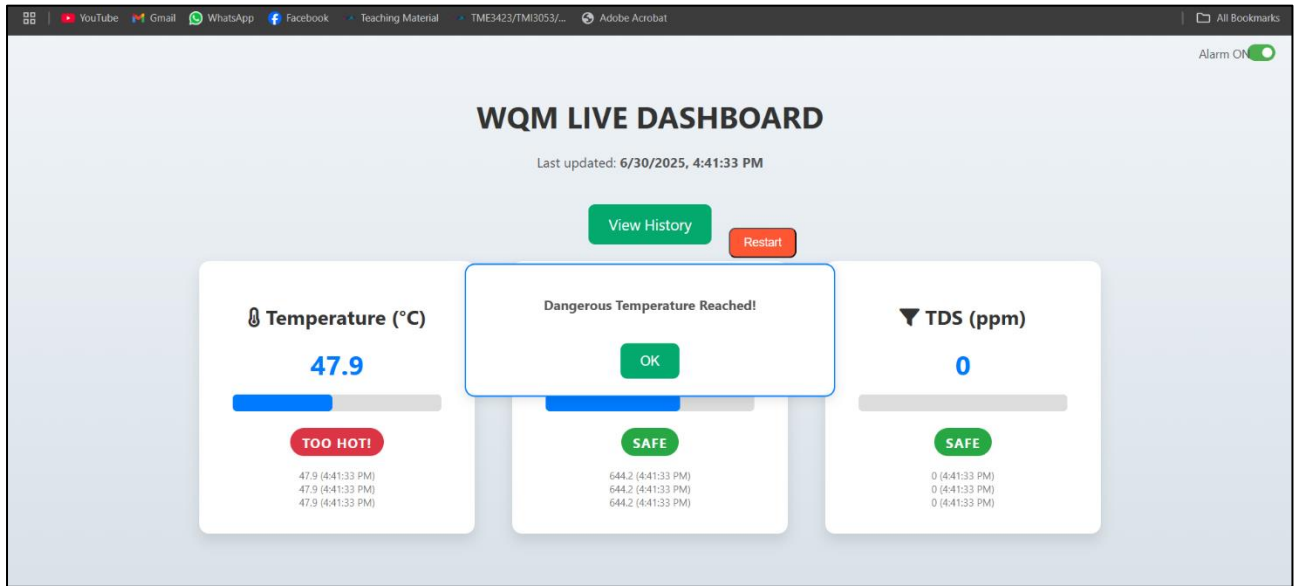


Figure 4.14 Notification for Alarm Trigger

Figure 4.14 shows a notification will pop up and an alarm will sound a warning if the temperature reach more than 40 celcius.

4.3.4 Data Handling

The screenshot shows a Microsoft Excel spreadsheet with a CSV file named "data_2025_06_weekly.csv". The spreadsheet has the following columns: id, temperature, turbidity, tds, and timestamp. The data is organized into two groups of readings, one for June 24th and one for June 26th, 2025.

| id | temperature | turbidity | tds | timestamp |
|------|-------------|-----------|-----|------------------|
| 6850 | 31 | 140 | 420 | 24/06/2025 8:00 |
| 7032 | 29.7 | 101 | 340 | 24/06/2025 8:00 |
| 7033 | 29.8 | 98 | 335 | 24/06/2025 9:00 |
| 7034 | 30.2 | 99 | 345 | 24/06/2025 10:00 |
| 7035 | 29.4 | 102 | 333 | 24/06/2025 11:00 |
| 7036 | 30.1 | 97 | 336 | 24/06/2025 12:00 |
| 7037 | 28.9 | 95 | 320 | 24/06/2025 13:00 |
| 7038 | 29.6 | 103 | 338 | 24/06/2025 14:00 |
| 7039 | 30.3 | 100 | 342 | 24/06/2025 15:00 |
| 7040 | 29.5 | 96 | 330 | 24/06/2025 16:00 |
| 7041 | 30.5 | 99 | 349 | 24/06/2025 17:00 |
| 1226 | 29.6 | 580.6 | 0 | 26/06/2025 3:29 |
| 1227 | 29.5 | 590.4 | 0 | 26/06/2025 3:29 |
| 1228 | 29.6 | 575.8 | 0 | 26/06/2025 3:29 |
| 1229 | 29.6 | 580.6 | 0 | 26/06/2025 3:29 |
| 1230 | 29.5 | 590.4 | 0 | 26/06/2025 3:29 |
| 1231 | 29.6 | 580.6 | 0 | 26/06/2025 3:29 |
| 1232 | 29.5 | 575.8 | 0 | 26/06/2025 3:30 |
| 1233 | 29.5 | 585.5 | 0 | 26/06/2025 3:30 |
| 1234 | 29.5 | 590.4 | 0 | 26/06/2025 3:30 |
| 1235 | 29.5 | 619.7 | 0 | 26/06/2025 3:30 |
| 1236 | 29.5 | 585.5 | 0 | 26/06/2025 3:30 |
| 1237 | 29.5 | 580.6 | 0 | 26/06/2025 3:30 |
| 1238 | 29.5 | 580.6 | 0 | 26/06/2025 3:30 |
| 1239 | 29.5 | 585.5 | 0 | 26/06/2025 3:30 |
| 1240 | 29.5 | 590.4 | 0 | 26/06/2025 3:30 |
| 1241 | 29.5 | 580.6 | 0 | 26/06/2025 3:30 |
| 1242 | 29.5 | 575.8 | 0 | 26/06/2025 3:31 |

Figure 4.15 CSV File for Data Exported

Figure 4.15 shows the example of data after being exported to csv file from the database. A scheduled export technique is used by the system to avoid ongoing database overload. Every week, using Windows Task Scheduler, a script called exports_weekly.php to exports all data older than seven days into a.csv file and stores it in the backup folder. The script removes the previous data from the MySQL database following a successful export. This guarantees effective data handling, particularly when sensors gather data continuously.

4.4 Prototype

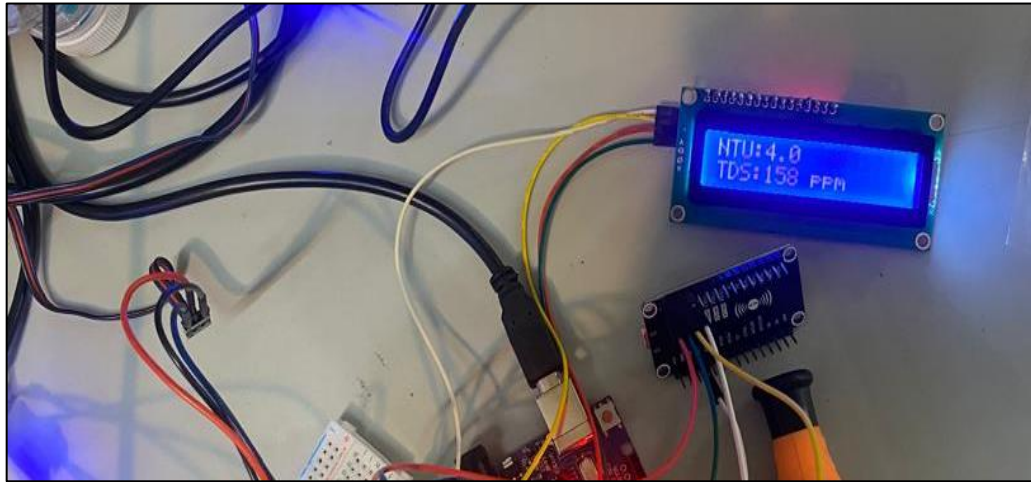


Figure 4.16 LCD Example Reading for TDS meter and Turbidity Sensor

Figure shows the prototype which is the reading of the sensors on the LCD. The Arduino uses the LiquidCrystal_I2C library to show this data on the LCD, which makes it easier to communicate with the LCD module using the I2C protocol. Only two communication lines are needed for this configuration, and they are linked to the Arduino UNO's A4 and A5 pins. SDA is used for data, and SCL is used for clock. Simple functions like lcd.setCursor() to position the text and lcd.print() to display the sensor readings are used by the Arduino to send formatted text to the screen after setting the LCD in the code.

4.5 System Modules

4.5.1 Database Module

The database module is responsible for storing water quality data such as turbidity, temperature, and total dissolved solids. It uses a MySQL database hosted locally via

XAMPP. The data is inserted through PHP scripts and can be queried by the web dashboard for both real time and historical analysis. The 'readings' table includes fields for each parameter and a timestamp.

4.5.2 Web Based Module

This module serves as the front-end interface for users to monitor real-time water quality data. It is built using HTML, CSS, JavaScript, and PHP. The dashboard displays the latest sensor readings. It interacts directly with the database module to fetch and display the most current data.

4.5.3 Microcontroller Integration Module

The microcontroller module, based on Arduino UNO, collects data from connected sensors and processes the values into readable units. It then sends this data to the web server using the ESP8266 Wi-Fi module. It also displays current values on an LCD using I2C protocol.

4.5.4 Sensor Data Acquisition Module

Each sensor (turbidity, temperature, and TDS) is part of the data acquisition module. These sensors read analog or digital signals from the water and pass the data to the Arduino UNO. The code ensures periodic and synchronized reading to avoid data collision or false results

4.6 Snippet Code Implementation

This section provides an overview and explanation of the main source code modules developed for the smart water quality monitoring system. The code is split between Arduino for sensor reading, ESP8266 for data transmission, and PHP scripts for data storage and dashboard display.

4.6.1 Arduino Sensor Reading

The Arduino UNO collects data from turbidity, TDS, and temperature sensors. It then displays the values on a 16x2 LCD and sends the data via Serial to ESP8266. Below is a snippet code:

```
Serial.print(tdsValue, 0);  
Serial.print(",");
```

```
Serial.print(turbidity, 1);  
Serial.print(",");  
Serial.println(temperature, 1);
```

Explanation:

The analogRead() reads the analog sensor value.
The voltage is calculated and converted to a units.
Data is sent in Comma-Separated Values (CSV) format over Serial for ESP8266 to capture.

4.6.2 ESP8266 for data transmission

The ESP8266 listens to Serial input from Arduino and uploads the data to the server via HTTP GET to insert.php.

```
String url = String(server) + "&tds=" + tds + "&turbidity=" + turb + "&temp=" + temp;  
http.begin(client, url);  
http.GET();
```

Explanation:

Constructs a full URL using sensor data.
This code shows the request is send to insert.php which stores the data in the Mysql databases.

4.6.3 PHP scripts

a) insert.php – Receives data from ESP8266 and stores it in the database.

```
$tds = $_GET['tds'];  
$turbidity = $_GET['turbidity'];  
$temp = $_GET['temp'];  
  
$sql = "INSERT INTO readings (tds, turbidity, temperature) VALUES ('$tds',  
'$turbidity', '$temp')"; // This line extract the parameters passed in the URL by the  
ESP8266.
```

b) `get_latest.php`- Retrieves the latest reading as JSON.

```
$sql = "SELECT * FROM readings ORDER BY timestamp DESC LIMIT 1";  
$row = $result->fetch_assoc();  
echo json_encode($row); // this code retrieves the most recent reading based on the  
timestamp.
```

c) `dashboard.php`- Displays the latest reading on the dashboard using PHP.

```
//javascript  
  
fetch('get_latest.php')  
  
  .then(res => res.json())  
  
  .then(data => {  
  
    document.getElementById('temp').innerText = data.temperature;  
  
    ...  
  
  }); // This code uses java script fetch() API to call get_latest.php every 5 seconds.
```

4.7 Summary

In this section, the implementation and testing steps have been completed and explained. In order to enable interaction between the functions, the system has been built and integrated. The system completed a number of testing processes for assessment. This testing process is important since it allows us to assess a system and make sure it has high-quality functions available.

5 Chapter 5: Software Testing

5.1 Overview

Software testing is a critical phase in the system development life cycle to ensure that the developed system functions correctly and meets its intended requirements. For this IoT based Water Quality Monitoring System, testing was performed after the development process was completed. The primary aim of testing is to validate whether the sensor readings, data transmission, storage, and display functionalities work as expected. This chapter discusses the system testing approach used, including both functional and nonfunctional testing.

5.2 System Testing

System testing is the process of assessing the system as a whole. It confirms that the integrated system satisfies the necessary specifications. This type of testing is categorised under the black box testing technique, which tests the inputs and outputs but does not analyse the internal logic. Both functional and nonfunctional components were tested in this project to guarantee the overall accuracy, stability, and usability of the system.

5.3 Functional Testing

Functional testing verifies the features and functionalities of the IoT water quality monitoring system. It ensures that all key modules, such as sensor integration, data transmission, database insertion, and LCD output, operate according to the system requirements. Table 5.1 show the functional testing.

Table 5.1 Functional Testing

| Functions | Expected Result | Success | Fail |
|------------------|---|---------|------|
| TDS sensor | TDS sensor can detect and send total dissolved solids (ppm) | / | |
| Turbidity sensor | Turbidity sensor can detect and send turbidity level (NTU) | / | |

| | | | |
|---------------------------|--|---|--|
| Temperature sensor | Temperature sensor can detect and send water temperature (Celsius) | / | |
| ESP8266 Data Transmission | Sensor data is successfully sent to the web server via Wi-Fi | / | |
| Database Storage | Sensor readings are stored in MySQL database | / | |
| LCD Display | Sensor readings are displayed in real time on the LCD | / | |
| Weekly Export & Cleanup | Data older than 7 days exported to .csv and removed from DB | / | |
| Temperature Alarm Trigger | Alarm plays and notifications appear when temp exceeds 40°C | / | |
| Alarm Toggle Function | Alarm can be turned off/on via dashboard toggle switch | / | |
| Restart Button | Clears all sensor readings and reloads new data after delay | / | |
| Previous Readings Log | Displays last 3 readings with time for each sensor | / | |

5.4 Non-Functional Testing

Nonfunctional testing ensures aspects of performance, usability, and reliability. In this system, testing was performed to verify data refresh rate, real time updates on the LCD, and consistency of database entries. It was also confirmed that the system maintained stable communication between Arduino and ESP8266 without crashing or data loss during extended periods. The system was tested in a controlled environment with sensors placed in regular water. Features like alarm toggling, restart action, and past reading visibility were validated under normal and extreme input conditions.

Non-functional testing focuses on evaluating the usability, aesthetics, and user satisfaction of the Smart Water Quality Monitoring System based on IoT. To perform this, a usability test was conducted involving 15 randomly selected respondents from different faculties. They interacted with the prototype and responded to a questionnaire using a 5-point scale (1: Completely Disagree to 5: Completely Agree). (Refer Appendices for Usablility Questionnaire). Figure 5.2 shows the usability test.

The evaluated items included:

Q1: Attractiveness and overall design of the system

Q2: System functionality (responsive and working features)

Q3: Accuracy of sensor readings (total dissolved solids, turbidity, temperature)

Q4: Ease of testing water using the sensors

Q5: Ease of reading data on the LCD

Q6: Clarity of system's purpose

Q7: Overall satisfaction with using the system

Table 5.2 Usability Test

| Respondent | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|------------|------|-----|-----|-----|-----|------|------|
| R1 | 3 | 4 | 5 | 4 | 4 | 5 | 4 |
| R2 | 2 | 3 | 3 | 3 | 3 | 3 | 2 |
| R3 | 4 | 4 | 5 | 4 | 4 | 5 | 4 |
| R4 | 3 | 5 | 5 | 5 | 5 | 5 | 5 |
| R5 | 3 | 5 | 4 | 4 | 5 | 5 | 4 |
| R6 | 3 | 4 | 4 | 4 | 5 | 4 | 4 |
| R7 | 3 | 3 | 4 | 4 | 4 | 3 | 3 |
| R8 | 2 | 4 | 3 | 4 | 3 | 3 | 3 |
| R9 | 3 | 4 | 2 | 4 | 4 | 5 | 4 |
| R10 | 3 | 4 | 3 | 4 | 4 | 4 | 3 |
| R11 | 3 | 5 | 3 | 4 | 4 | 5 | 4 |
| R12 | 3 | 4 | 4 | 4 | 3 | 4 | 4 |
| R13 | 4 | 4 | 5 | 5 | 5 | 5 | 5 |
| R14 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| R15 | 3 | 4 | 4 | 4 | 4 | 5 | 4 |
| Average | 2.93 | 4.0 | 3.8 | 4.0 | 4.0 | 4.27 | 3.73 |

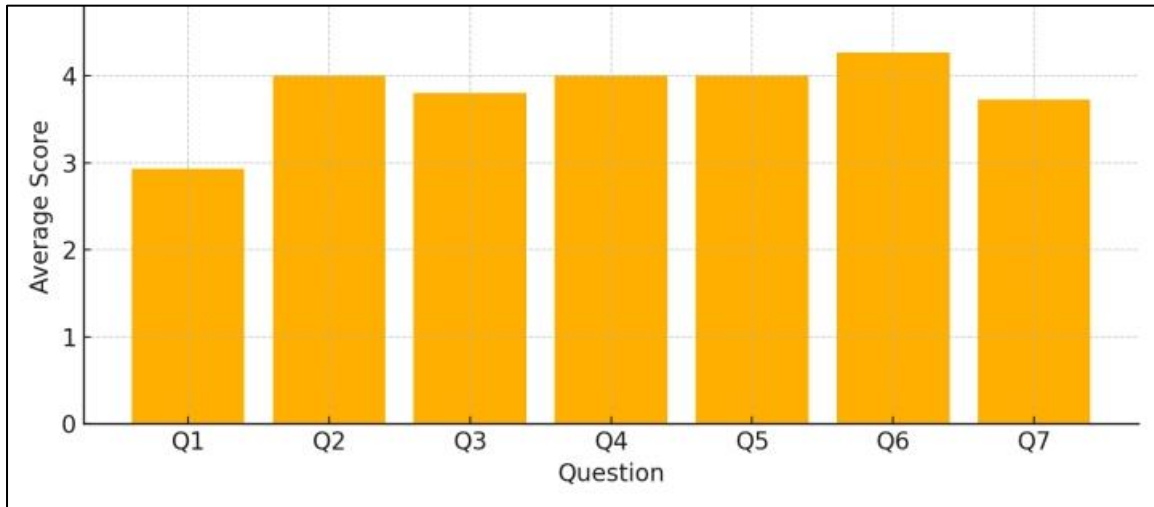


Figure 5.1 Average Usability Scores for Each Question

Figure 5.1 illustrates the average scores for each usability aspect. As seen, the system performs well in terms of sensor functionality and data readability (Q3, Q4, Q5), scoring above 4.0. However, the attractiveness and physical wiring setup (Q1) scored lower due to its perceived complexity. Users found the setup less user-friendly, especially those unfamiliar with electronics. The overall satisfaction (Q7) remains positive, indicating general approval despite some design challenges. These insights will be considered for future improvements.

5.5 Unit Testing

The table below lists the functional test cases used to verify that each component of the system performs as expected:

Table 5.3 Test Cases

| Test Case ID | Description | Expected Result | Actual Result | Status |
|--------------|--------------------------------|-----------------------------|---------------|--------|
| TC01 | Read TDS sensor on A2 | TDS value in ppm (0-2000) | 400 | Pass |
| TC02 | Read Turbidity sensor on A1 | NTU shown between 0-1000 | 679 | Pass |
| TC03 | Read Temperature sensor (D4) | Value in °C (20-40) | 27.5 | Pass |
| TC04 | Display all 4 readings on LCD | LCD shows values correctly | Yes | Pass |
| TC05 | ESP sends data via Serial | Serial sends correct format | 100.72.2.20 | Pass |
| TC06 | ESP connects to Wi-Fi | ESP shows connected | Connected | Pass |
| TC07 | ESP sends GET to insert.php | Data appears in DB | Yes | Pass |
| TC08 | get_latest.php returns data | JSON with latest reading | Yes | Pass |
| TC09 | dashboard2.php displays latest | Values shown on screen | Yes | Pass |

5.6 Integration Test Cases

The following integration tests ensure that the components work correctly when integrated together:

| Test ID | Scenario | Expected | Status |
|---------|--------------------------------|---|--------|
| INT01 | Arduino + ESP + insert.php | DB receives correct data | Pass |
| INT02 | Dashboard loads after ESP push | New values appear within 5 seconds | Pass |
| INT03 | Dashboard reloads correctly | No crash, no error | Pass |
| INT04 | Export script + Scheduler | Automatically backs up old data weekly via Task Scheduler | Pass |
| INT05 | Alarm integration | Alarm triggers only when enabled and temperature is high | Pass |
| INT06 | Restart button integration | Readings clear and refresh smoothly on button click. | Pass |

5.7 Summary

During testing, every important system module operated as planned. Real time data from the sensors was successfully stored in the MySQL database and shown on the LCD. Conducting user acceptability and stress testing to assess the system under greater pressures or wider installation scenarios are recommendations for future enhancement.

6 Chapter 6: Conclusion and Future Works

6.1 Overview

This chapter is the last chapter for Final Year Project outlines the achievements of the proposed project and the problems faced throughout the development of the system as well the potential for future works that can be done to the proposed system in order to produce a complete system that bring benefits to the user.

6.2 Achievements

A water quality monitoring system has been successfully developed after 6months to encounter the problem that has been addressed in Chapter 1 of the project. The system has also been developed based on the objectives of the project. The objectives and its corresponding achievements are shown in Table 6.1.

Table 6.1 Objectives and Achievements

| Objectives | Achievements |
|--|---|
| To design an efficient Internet of Things (IoT)-based water quality monitoring system that incorporates sensors to measure important water parameters (temperature, turbidity, total dissolved solids) | The model for the water quality monitoring system has been develop to examine water quality which are temperature, turbidity, total dissolved solids. |
| To develop a web-based system that enables real-time monitoring to water quality information. | The web-based system is manage to do for data visualization. |
| To evaluate the effectiveness and reliability of the water quality monitoring system | The sensors has shown a great effectiveness and reliability on the water quality based on the readings. |

Reason objectives:

- For the web dashboard, this project is hosted locally using XAMPP instead of deploying it to a live hosting environment. Hosting locally allowed to:
 - Quickly troubleshoot and debug issues in real time
 - Simulate realistic IoT data interactions in a controlled environment

6.3 Limitations and Problem Encountered

The problems that have been faced throughout the development process creating proposed system are obtaining data from the sensors. Besides that, despite having problems building proposed system, sometimes there was also problems in retrieving the data live to the dashboard due to the connection and sensitivity to the wiring. There is also occasional instability in Wi-Fi data transmission.

The turbidity sensor the is used in this project is an analog type, and it doesn't always return to 0 when taken out of water because some water drops or dirt might still be on the sensor. These leftover particles can affect the reading even if it's not in water. Also, the sensor doesn't give an exact NTU value because it's very sensitive to light, reflections, and small changes in the environment. Without proper calibration using clean water or a reference solution, the value may not be 100% accurate. So, instead of showing a perfect number, the sensor is more useful for checking whether the water is clear or dirty in general.

6.4 Future Work

For the future development of this project, several aspects can be considered to improve the project such as:

- Allowing the system to store and display the history of the sensor data in the dashboard. This can expand the system to be more reliable and significant and also makes it easier for the user to find the data.
- Include battery operated solar power for remote areas.
- Allowing the application of water quality monitoring to view the trend of water quality from time to time and develop the mobile application to allow the user to remotely monitor the water quality everywhere they can.
- Use more sensor for example pH sensor to check the the quality of water whether it is acids or alkali.

6.5 Summary

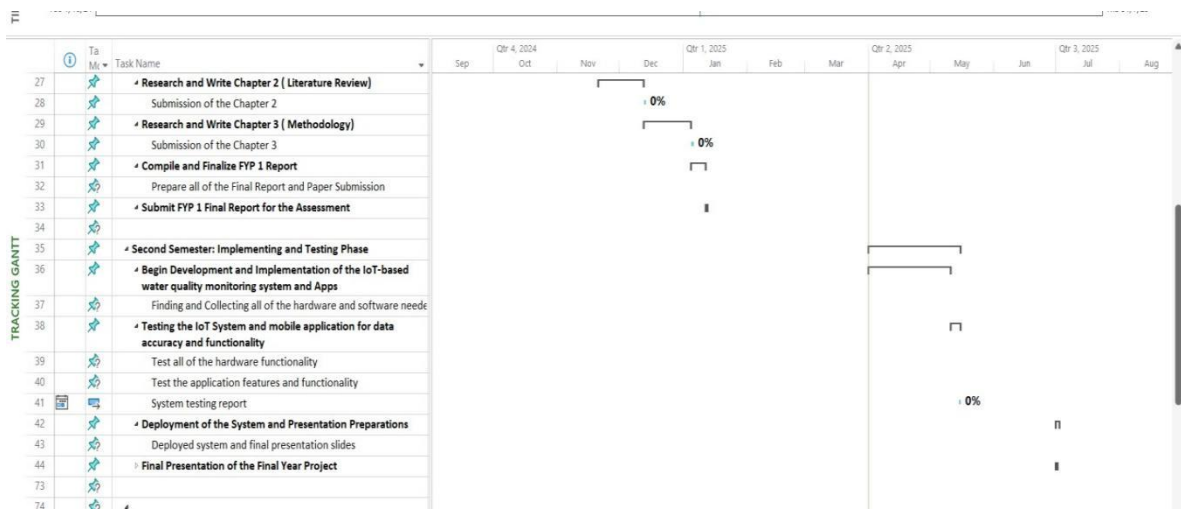
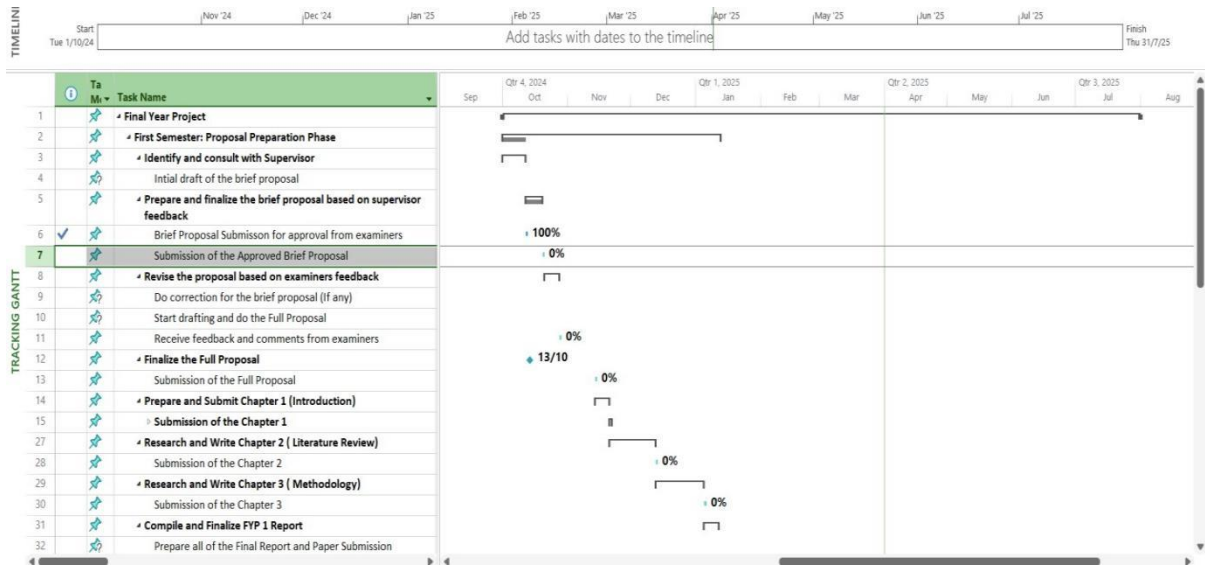
A Smart Water Quality Monitoring Based on IoT system is a system that offers solution to the users on their inquiries on the water quality by providing information of the level of water quality which are temperature, total dissolved solids and turbidity. By the existence of this system, it will help the people all around the world to check the quality of water for example lake and many more as water is essential for daily living.

References

- AlMetwally, S. A. H., Hassan, M. K., & Mourad, M. H. (2020). Real-time internet of things (IoT) based water quality management system. *Procedia CIRP*, *91*, 478-485.
- Air Selangor. (2020). Unscheduled water supply disruption in 7 regions starting at 6.00 PM today following raw water pollution. Retrieved January 21, 2021, from <https://www.airselangor.com/gangguan-bekalan-air-tidak-berjadual-di-7-wilayah-bermula-jam-600-petang-hari-ini-berikutan-pencemaran-sumber-air-mentah/>.
- Babatunde, A. (2024). A study on traditional water quality assessment methods. *Risk Assessment and Management Decisions*, *1*(1), 41-52.
- Danh, L. V. Q., Dung, D. V. M., Danh, T. H., & Ngon, N. C. (2020). Design and deployment of an IoT-based water quality monitoring system for aquaculture in Mekong Delta. *International Journal of Mechanical Engineering and Robotics Research*, *9*(8), 1170-1175.
- Department of Statistics Malaysia. (2024, October 17). The Population of Malaysia. <https://open.dosm.gov.my/dashboard/population>.
- Glasgow, H. B., Burkholder, J. M., Reed, R. E., Lewitus, A. J., & Kleinman, J. E. (2004). Real-time remote monitoring of water quality: A review of current applications, and advancements in sensor, telemetry, and computing technologies. *Journal of Experimental Marine Biology and Ecology*, *300*(1-2), 409-448.
- He, D., & Zhang, L. X. (2012, April). The water quality monitoring system based on WSN. In *2012 2nd international conference on consumer electronics, communications and networks (CECNet)* (pp. 3661-3664). IEEE.
- Korzeniewska, E., Zawiaślak, R., Przybył, S., Sarna, P., Bilska, A., & Mączka, M. (2023). Prototype of Data Collector from Textronic Sensors. *Sensors*, *23*(24), 9813.

- Loi, J. X., Chua, A. S. M., Rabuni, M. F., Tan, C. K., Lai, S. H., Takemura, Y., & Syutsubo, K. (2022). Water quality assessment and pollution threat to safe water supply for three river basins in Malaysia. *Science of The Total Environment*, 832, 155067.
- Nadu, T. (2020). An underground pipeline water quality monitoring using IoT devices. *European Journal of Molecular and Clinical Medicine*, 7, 2046-2054.
- Wu, Q., Liang, Y., Sun, Y., Zhang, C., & Liu, P. (2010, September). Application of GPRS technology in water quality monitoring system. In *2010 World Automation Congress* (pp. 7-11). IEEE.
- Zhang, J., Sheng, Y., Chen, W., Lin, H., Sun, G., & Guo, P. (2021). Design and analysis of a water quality monitoring data service platform. *Computational Materials Continua*, 66(1), 389-405.
- Zhang, Y., Deng, J., Qin, B., Zhu, G., Zhang, Y., Jeppesen, E., & Tong, Y. (2023). Importance and vulnerability of lakes and reservoirs supporting drinking water in China. *Fundamental Research*, 3(2), 265-273.
- Zulkifli, C. Z., Garfan, S., Talal, M., Alamoodi, A. H., Alamleh, A., Ahmaro, I. Y., ... & Chiang, H. H. (2022). IoT-based water monitoring systems: A systematic review. *Water*, 14(22), 3621.

Appendix A



Appendix B-Usability Text Questionnaire

Smart Water Quality Monitoring Based on IoT

Objectives: To evaluate the usability and functionality of the developed system based on user experience.

Section A: Respondent Information

1. Gender:

Male Female

2. Faculty:

3. Have you used any water monitoring system or similar system before?

Yes No

Section B: System Evaluation

Please indicate your level of agreement with the following statements. Circle the number that best represents your opinion. (1 = Strongly Disagree, 5 = Strongly Agree)

| No. | Statement | 1 Strongly Disagree | 2 Disagree | 3 Neutral | 4 Agree | 5 Strongly Agree |
|-----|--|---------------------------|---------------|--------------|------------|------------------------|
| 1 | The system is visually attractive and well-designed. | | | | | |
| 2 | The system features work as intended and are functional. | | | | | |

| | | | | | | |
|---|---|--|--|--|--|--|
| 3 | Sensor readings appear accurate and consistent. | | | | | |
| 4 | It is easy to test water samples using the system. | | | | | |
| 5 | It is easy to read the data on the LCD screen. | | | | | |
| 6 | The purpose of the system is clearly understood. | | | | | |
| 7 | I am satisfied with my experience using the system. | | | | | |

Additional Comments:
