WATER QUALITY MONITORING SYSTEM WITH ARDUINO AND K-NEAREST NEIGHBORS MODEL

BARANEESWAR A/L RAVI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

WATER QUALITY MONITORING SYSTEM WITH ARDUINO AND K-NEAREST NEIGHBORS MODEL

BARANEESWAR A/L RAVI

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Artificial Intelligence) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

WATER QUALITY MONITORING SYSTEM WITH ARDUINO AND K-NEAREST NEIGHBORS MODEL

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT: ________________________ Date :27/8/2024

(BARANEESWAR A/L RAVI)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Software Development) with Honours.

SUPERVISOR : ________________________ Date : 20/9/2024

(PROFESSOR TS. DR. BURHANUDDIN BIN MOHD ABOOBAIDER)

DEDICATION

I dedicate this report to my parents, Ravi Subramaniam and Jayamoney Arumugam. Their endless support, constant encouragement and love have been the cornerstone of my academic journey. Appa, your wisdom and guidance have inspired me to persevere and strive for excellence. Amma, your care and belief in me have provided the strength I needed to overcome every obstacle in my life.

Your sacrifices have not gone unnoticed. The countless hours spent nurturing my curiosity, the late nights you stayed up to help me with my studies, and the many ways you went out of your way to ensure I had the opportunities to succeed, all of these acts of love have shaped who I am today. Your patience, resilience and faith in my potential have taught me the true meaning of perseverance and hard work.

A heartfelt thank you to my mentors, and lecturers, especially Professor Ts. Dr. Burhanuddin Bin Mohd Aboobaider whose wisdom, guidance and expertise have shaped the course of this project. Your dedication to imparting knowledge and your commitment to nurturing my intellectual growth have been truly inspiring. The insights and skills I have gained from you will continue to influence my work and thinking for years to come.

Lastly, I dedicate this report to countless individuals whose hard work and dedication to their fields laid the foundation for my own learning. Your efforts have created the foundation upon which this work is built. This achievement is a testament to the collective support, wisdom and encouragement of many. I a m profoundly grateful to each and every one of you.

ACKNOWLEDGEMENTS

This report would not have been possible without the support, guidance and assistance of may individuals. I would like to take this opportunity to express my gratitude to everyone who contributed to the completion of this work.

First and foremost, I would like to thank my advisor, Professor Ts. Dr. Burhanuddin Bin Mohdaboobaider, for their invaluable guidance, insightful feedback, and unwavering support throughout this project. Your expertise and dedication have been instrumental in shaping the direction and quality of this report. I am deeply grateful for your mentorship and the countless hours you have invested in my success.

A special thank you to Universiti Teknikal Malaysia Melaka for providing the resources and support necessary to conduct this research. The access to facilities, funding, and academic resources has been crucial in enabling me to complete this project.

I am deeply appreciative of my peers and faculty members of the Faculty of Information and Communication Technology who have provided valuable feedback, shared their knowledge, and offered their support throughout this journey. Your camaraderie and collaboration have made this process more enjoyable and intellectually stimulating.

Lastly, I acknowledge and appreciate myself. I thank myself for the dedication, hard work, and resilience I have shown throughout this endeavor. I thank myself for the commitment to continuous improvement, for striving to give more than I receive, and for always staying true to myself.

ABSTRACT

The increasing demand for potable water necessitates advanced monitoring systems to ensure water quality. Monitoring water quality is one of the major objectives of this project where an ESP32 microcontroller which incorporates temperature, TDS, and pH Sensors is used. The first and foremost aim is to use machine learning for classifying the given water samples as either potable or not potable; this has been focused to avoid the issue of missing data values for the datasets and works to improve the prediction capability. Regarding the scope of the system, it identifies water samples by employing the k-Nearest Neighbors (kNN) algorithm the system uses real-time sensor reading stored in Firebase. Samples for data collection involve the measurements of various sensors that include temperature, Ph, and turblidity and makes sure we have a large data sample including those from potable and non-portable water sources. The preprocessing step involves cleaning to address cases of missing values and feature extraction to obtain the desirable attributes. Next, the preprocessed data is split into training and testing datasets, where the kNN model is trained with training data and employed to determine accuracy using the testing data. When the model has been proven to perform well in the simulation, it is then utilized in a real environment for water quality monitoring. It is important to have a clean and intuitive user interface of a mobile application where real-time data are shown and kNN will be used to make instant data retrieval with the help of Firebase solution. The system also includes a constant monitor and data acquisition of predefined sensors to detect water quality parameters that fall out of range. The system has been extensively tested and validated to confirm its measurability, with ongoing monitoring incorporated to handle issues that may appear. This paper's objective lies in offering a sustainable and efficient solution for the periodical monitoring of water quality, facilitating solutions for providing people with safe drinking water. The dataset consists of 1,000 samples acquired from publicly available water quality datasets on the internet, ensuring a comprehensive range of potable and non-potable water scenarios. While real-time data collection is implemented using the ESP32 microcontroller and sensors, future development plans include expanding the use of IoT for continuous data acquisition and integrating a larger, more dynamic database. This will enable more robust real-time monitoring, as well as automatic alerts for potential water quality issues.

ABSTRAK

Permintaan yang semakin meningkat terhadap air yang boleh diminum memerlukan sistem pemantauan yang maju untuk memastikan kualiti air. Pemantauan kualiti air merupakan salah satu objektif utama dalam projek ini di mana mikropengawal ESP32 yang menggabungkan sensor suhu, TDS, dan pH digunakan. Matlamat pertama adalah menggunakan pembelajaran mesin untuk mengklasifikasikan sampel air yang diberikan sama ada boleh diminum atau tidak; ini difokuskan untuk mengelakkan masalah nilai data yang hilang dalam set data dan meningkatkan keupayaan ramalan. Berkaitan dengan skop sistem, ia mengenal pasti sampel air dengan menggunakan algoritma k-Nearest Neighbors (kNN) di mana sistem menggunakan bacaan sensor masa nyata yang disimpan di Firebase. Sampel untuk pengumpulan data melibatkan pengukuran daripada pelbagai sensor yang merangkumi suhu, pH, dan kekeruhan serta memastikan kami mempunyai sampel data yang besar termasuk yang dari sumber air boleh diminum dan tidak boleh diminum. Langkah prapemprosesan melibatkan pembersihan untuk menangani kes nilai yang hilang dan pengekstrakan ciri untuk memperoleh atribut yang diingini. Seterusnya, data yang dipraproses akan dibahagikan kepada set latihan dan ujian, di mana model kNN dilatih dengan data latihan dan digunakan untuk menentukan ketepatan menggunakan data ujian. Apabila model telah terbukti berprestasi baik dalam simulasi, ia kemudian digunakan dalam persekitaran sebenar untuk pemantauan kualiti air. Adalah penting untuk mempunyai antara muka pengguna mudah alih yang bersih dan intuitif di mana data masa nyata dipaparkan, dan kNN akan digunakan untuk membuat pengambilan data segera dengan bantuan penyelesaian Firebase. Sistem ini juga merangkumi pemantauan dan pengambilan data berterusan daripada sensor yang telah ditetapkan untuk mengesan parameter kualiti air yang berada di luar julat yang boleh diterima. Sistem ini telah diuji dan disahkan secara meluas untuk mengesahkan keukurannya, dengan pemantauan berterusan yang disertakan untuk menangani isu-isu yang mungkin timbul. Objektif kertas ini adalah untuk menawarkan penyelesaian yang mampan dan cekap bagi pemantauan berkala kualiti air, memudahkan penyelesaian untuk menyediakan air minuman yang selamat kepada orang ramai. Set data terdiri daripada 1,000 sampel yang diperoleh daripada set data kualiti air yang tersedia secara dalam talian, memastikan rangkaian yang komprehensif bagi senario air boleh diminum dan tidak boleh diminum. Walaupun pengumpulan data masa nyata dilaksanakan menggunakan mikropengawal ESP32 dan sensor, rancangan pembangunan masa depan termasuk memperluaskan penggunaan IoT untuk pengambilan data berterusan dan mengintegrasikan pangkalan data yang lebih besar dan dinamik. Ini akan membolehkan pemantauan masa nyata yang lebih mantap serta amaran automatik untuk masalah kualiti air yang berpotensi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I: INTRODUCTION

1.1 Introduction

Monitoring and conservation of Water Quality In a world reeling under the effect of environmental deterioration, water quality monitoring and conservation take on paramount importance. Urbanization and industrialization can increase the risk of water pollution thereby threatening ecosystems and public health. To tackle this issue, solutions worked and affordable are being provided using cutting-edge technologies. Arduino is a complete electronics development platform that is very popular due to its simplicity and versatility in which can be used for the construction of numerous items, and one of these developments may be able to applied as an Water Quality Monitoring System. This particular system has sensors to measure a number of different aspects of water quality, such as temperature, turbidity, and pH level. It subsequently gives real-time data on not meeting benchmark criteria early enough to be able approach it. This, thus ensures an affordable and custom made (as Water Quality Monitoring System is made with Arduino) two-parts unit of the water quality monitoring system.

1.2 Problem Statement

i. Increasing Water Contamination Due to Urbanization and Industrialization

Urbanisation and industrialisation have been increasing worldwide leading to significant raise in urban areas as well as industries which has in turn led to water pollution posing a big problem. This expanding concern of rising water contamination calls for urgent deployment of efficient mechanisms of monitoring the quality of water to secure better the environment and health of people living there.

ii. Limitations of Current Water Quality Monitoring Techniques

The existing techniques used in assessing the quality of water are usually expensive, inaccessible, and cannot provide real-time data mostly used by industries and other broad platforms such as agriculture which may not be accurate when consumers want to use that particular water or community should consume this treated supply. These limitations impair early detection and prevention of contamination, thus making it quite difficult to manage or mitigate pollution effectively.

iii. Need for an Affordable and Accessible Monitoring Solution

There is an acute dearth regarding full-fledged low-cost easily installed water quality monitor around today. One plausible way out can be an Arduino-based approach that offers a chance being able to monitor continuously, uncomplicated interface, rapid environmental threat identification among many others.

1.3 Objectives

i. To ensure continuous monitoring by maintaining both historical and real-time water quality data without any gaps.

ii. To develop a user-friendly interface through which individuals can understand it and take appropriate actions as per the water quality messages.

iii. To provide users with actionable insights for informed decision-making and environmental management.

1.4 Scope

i. Target User

This project is designed for the eco-conscious users, small organizations, colleges and universities and potential business clients. They all need to have an assurance of reliable water quality information updated in real time to guarantee safety and sustainability of water resources.

ii. Module to be developed

i. Hardware Module

The Hardware Module consists of the physical components necessary for measuring water quality parameters. This includes incorporating different sensors such as temperature sensors, solid metal sensors and pH sensors with Arduino platform; this module will also cover circuit board design, power supply design and communication interfaces that can transfer data to the central system; finally, it will take into account its development through being robust and durable in different environmental conditions.

ii. Artificial Intelligence Module

The AI module leverages machine learning algorithms to process and analyze the data collected by these sensors themselves. The module will recognize patterns, identify anomalies, predict if a water is potable or not in advance before they get worse. The artificial intelligence (AI) module can process huge amounts of data thereby enhancing its capacity to provide real-time insights and recommendations that foster proactive management of environment as well as quick responses to emerging threats.

iii. System Module

The System Module is the main framework which brings together the hardware and AI components into a compact water quality monitoring solution. This module administers information management, acquisition, treatment and visualization. It involves creation of user-friendly interfaces like dashboards and mobile apps that make data easily understandable and actionable. Accordingly, the system module guarantees a smooth flow of information between the hardware and AI as well as end-users with real time updates and alerts for effective water quality monitoring.

1.5 Project Significance

The motivation for this project is to develop a low-cost, user-friendly, and effective water quality monitoring system to significantly enhance water quality management and public health. By leveraging the accessible and affordable Arduino platform, this system addresses the limitations of traditional methods, such as high costs and lack of real-time data. The project aims to make continuous water quality monitoring accessible to a broader range of users, including communities, local governments, and organizations in developing countries, thereby improving public health by preventing waterborne diseases. Additionally, the system promotes sustainable development by protecting water resources and preserving ecosystems. The integration of artificial intelligence offers advanced data analysis and actionable insights, empowering users to make informed decisions and respond promptly to contamination events. This project has the potential to revolutionize water quality monitoring, leading to significant improvements in water quality, human health, and environmental sustainability.

1.6 Expected Output

The expected outcome for this project is the creation of a comprehensive water quality monitoring system, meticulously designed to offer a multifaceted approach towards ensuring the quality and safety of water resources. At its core, the hardware module will constitute a robust framework integrating an array of sophisticated sensors, calibrated to capture an extensive spectrum of water parameters. These sensors, ranging from those measuring temperature differentials to those assessing solids levels and pH fluctuations, will collectively provide a granular and real-time depiction of the water's quality. This wealth of data will then be harnessed by the artificial intelligence module, which serves as the analytical powerhouse of the system. Employing advanced machine learning algorithms, this module will not only process the influx of data but also discern patterns, identify anomalies, and forecast potential deviations from expected water quality norms. This predictive capability stands as a cornerstone in the system's ability to proactively anticipate and mitigate emerging contamination threats, thereby ensuring the timely implementation of preventive measures and safeguarding the well-being of ecosystems and human populations reliant on these water bodies.

In essence, the expected output of this project transcends the mere development of a technological artifact; it embodies a paradigm shift towards a more holistic and proactive approach to water quality management. By amalgamating cutting-edge technology with the ethos of environmental stewardship, this system not only revolutionizes water quality monitoring but also serves as a catalyst for sustainable development. In doing so, it underscores the pivotal role of technology in addressing pressing environmental challenges, fostering resilience, and ushering in an era of heightened environmental consciousness and responsibility.

1.7 Conclusion

The Water Quality Monitoring System using Arduino introduces an affordable, user-friendly, and cutting-edge monitoring solution to solve the urgent problem of water contamination. With an eye toward surpassing the constraints of current approaches, the project makes use of the adaptability of the open-source electronics platform Arduino. The system has sensors that measure temperature, turbidity, and pH levels, among other important water factors, in real time. The project intends to preserve valuable water resources by offering a prompt response to deviations from set water quality requirements through continuous monitoring. This creative solution is a big step toward building an extensive and easily available surveillance system to protect against the growing risks associated with water contamination in our quickly changing industrial and urban environments.

CHAPTER II: LITERATURE REVIEW AND METHODOLOGY

2.1 Introduction

This chapter will discuss the literature review for the existing system that are related to the proposed project and the technique that has been used in this project is the Machine Learning (ML) technique which is the subset of Artificial Intelligence (AI) field. The project methodology, project requirements, and project schedule and milestones will be stated with the details.

2.2 Facts and Findings

This part will discuss the domains regarding this project which will help in understanding the core value of this project and the completion of this project.

2.2.1 Domain

The domain related to this project is the realm of water quality assessment and potability determination using advanced machine learning techniques. The project focuses on developing a robust water quality monitoring system capable of assessing whether water is potable or not based on sensor readings and data analysis. Water quality is a critical concern globally, and ensuring access to safe drinking water is paramount for public health and environmental sustainability. Traditional methods of water quality assessment often rely on laborious and time-consuming laboratory tests, making real-time monitoring and decision-making challenging. Therefore, the project aims to leverage machine learning algorithms, specifically the k-Nearest Neighbors (kNN) algorithm, to provide rapid and accurate assessments of water potability. By analyzing sensor data collected from various parameters such as pH levels, turbidity, and dissolved solids, the system determines the suitability of water for drinking purposes.

The kNN algorithm, known for its simplicity and effectiveness in classification tasks, will be employed to classify water samples as potable or non-potable based on similarity to existing labeled data in the dataset. This approach enables quick identification of potential contamination or quality issues, allowing for timely intervention to ensure safe drinking water supply. The project's domain encompasses the intersection of environmental science, sensor technology, and machine learning, with the ultimate goal of enhancing water quality monitoring and ensuring access to safe drinking water for communities worldwide.

2.2.2 Existing System

Several studies have explored the state of water quality monitoring systems, investigating various techniques and methodologies to ensure the safety and sustainability of water resources. A. Juna et al. (2022) proposed an innovative approach to water quality prediction, employing a combination of a K-Nearest Neighbor (KNN) imputer and a Multilayer Perceptron (MLP). Their study addresses a critical challenge in water quality prediction: the presence of missing values in datasets, which can significantly reduce the accuracy of predictive models.

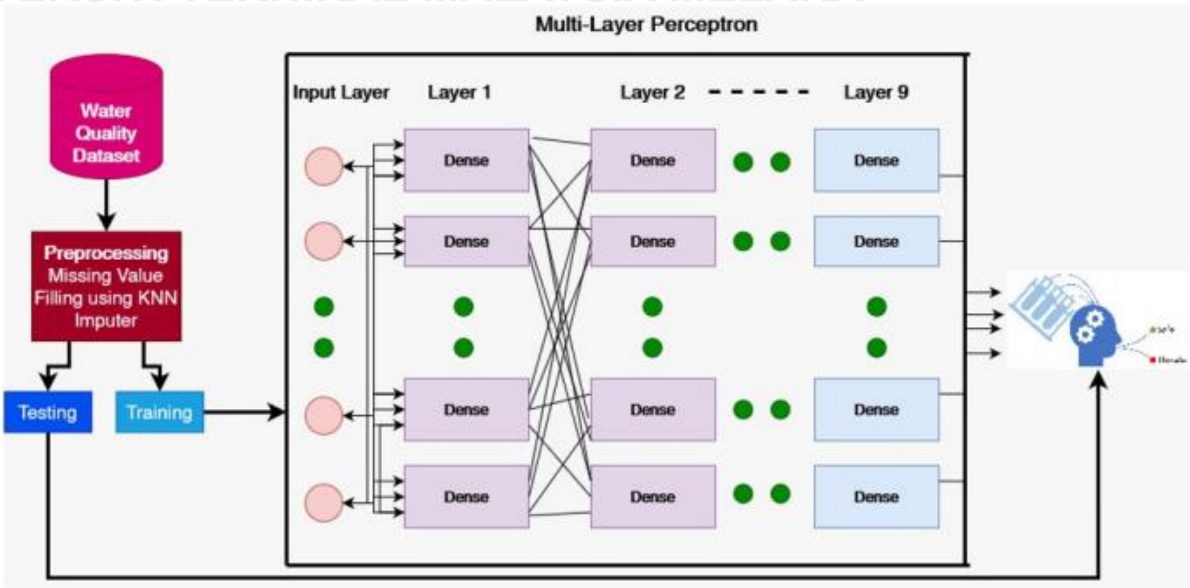

Figure 2.1 Architecture of the existing approach

The system architecture involves using the KNN imputer to handle missing values in the water quality datasets before applying the nine-layer MLP for prediction. This dual approach is designed to enhance the robustness and accuracy of water quality predictions. Juna et al. conducted extensive experiments to compare the performance of their proposed model with seven other machine learning algorithms, including scenarios where missing values were either deleted or imputed using KNN.

To evaluate the effectiveness of their models, Juna et al. utilized a comprehensive dataset obtained from Kaggle. They employed metrics such as accuracy, precision, recall, F1 score, and Area Under the Curve (AUC) to assess model performance. The outcomes of their study demonstrated that the nine-layer MLP model with the KNN imputer achieved an outstanding accuracy of 0.99 in predicting water quality, significantly surpassing other machine learning techniques.

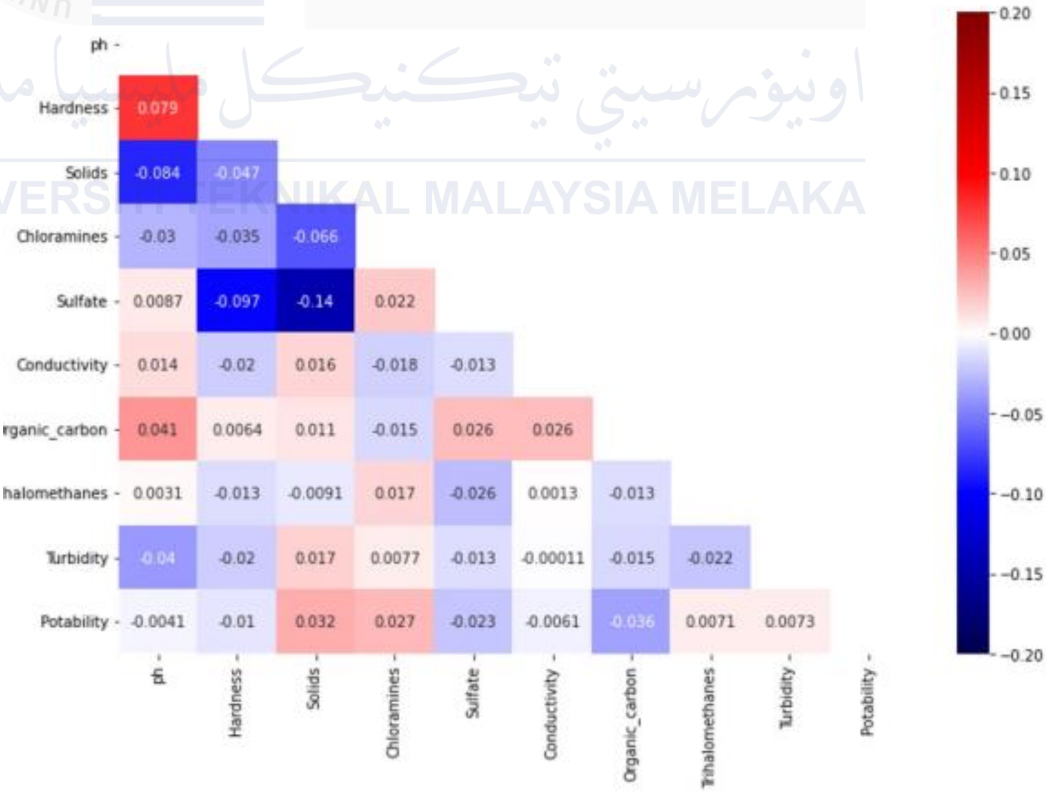

Figure 2.2 Correlation heatmap of features.

The research highlights that traditional water quality prediction approaches often lack the desired accuracy due to the issue of missing values. By using the KNN imputer, Juna et al. effectively addressed this problem, showing that imputing missing values rather than deleting them can lead to substantial improvements in predictive performance. Their results suggest that the proposed approach not only enhances accuracy but also ensures the reliability of water quality predictions.

Feature	Description
pH	pH of water (0 to 14).
Hardness	Capacity of water to precipitate soap in mg/L.
Solids	Total dissolved solids in ppm.
Chloramines	Amount of chloramines in ppm.
Sulfate	Amount of sulfates dissolved in mg/L.
Conductivity	Electrical conductivity of water in μS/cm.
Organic_carbon	Amount of organic carbon in ppm.
Trihalomethanes	Amount of trihalomethanes in μg/L.
Turbidity	Measure of light-emitting property of water in NTU.
Potability	Indicates if water is safe for human consumption. Potable, 1, and not potable, 0.

Figure 2.3 Description of the dataset used in this study

The study by Juna et al. underscores the potential of combining machine learning techniques to tackle the challenges in water quality monitoring systems. Their approach demonstrates the value of using advanced imputation methods to handle incomplete datasets, paving the way for more accurate and efficient water quality assessment tools.

In conclusion, the research by A. Juna et al. (2022) makes a significant contribution to the field of water quality prediction by introducing a robust model that combines a KNN imputer and a multilayer perceptron. This innovative approach addresses the common issue of missing values in datasets, leading to highly accurate predictions. The findings of this study have important implications for environmental monitoring and public health, offering a scalable and reliable solution for ensuring access to safe drinking water. The combination of the KNN imputer with the MLP model sets a new benchmark in predictive accuracy, demonstrating the potential of machine learning techniques in revolutionizing water quality monitoring systems.

2.2.3 Technique

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that empowers computers to learn from data without explicit programming. In the context of water quality monitoring, ML algorithms play a crucial role in analyzing sensor data and identifying potential contamination or quality issues. Figure 2.1 illustrates the different pathways of ML, including supervised learning, unsupervised learning, and reinforcement learning.

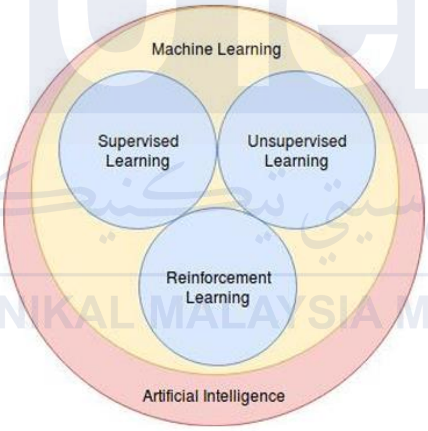

Figure 2.4 Field of Artificial Intelligence

One of the primary ML paradigms utilized in this project is supervised learning, which relies on labeled datasets to train models. Supervised learning algorithms, such as k-Nearest Neighbors (kNN), classify data based on known outcomes. For water quality monitoring, kNN is particularly suitable for detecting potable water based on sensor readings.

Figure 2.5 Supervised Learning

Next, kNN operates by assigning data points to classes based on the majority class among their nearest neighbors. This intuitive approach is effective, especially when dealing with datasets lacking clear class separation. The choice of kNN for this project is supported by its simplicity and effectiveness in classification tasks.

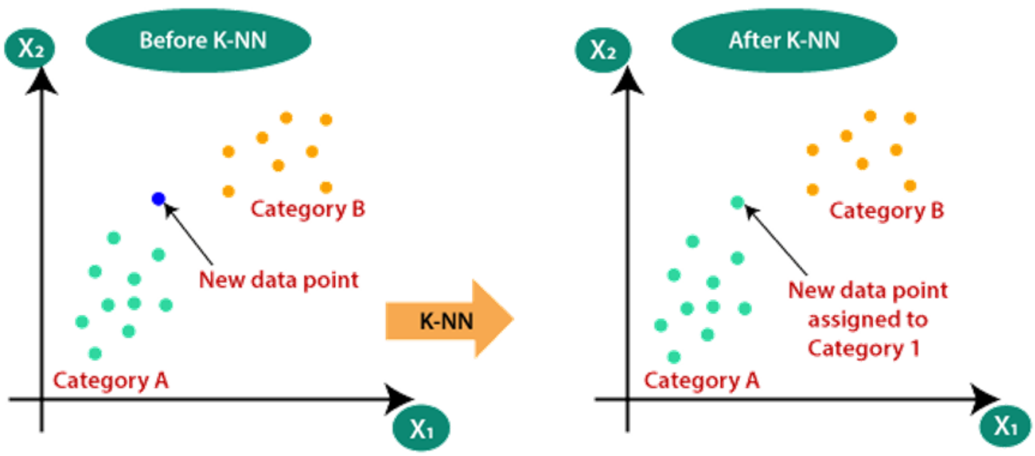

Figure 2.6 KNN Model

In addition to kNN, other ML models are considered for comparison, including Decision Trees, Random Forest, Support Vector Machines (SVM), and Logistic Regression. Each of these models possesses unique attributes and capabilities suitable for water quality monitoring tasks. Decision Trees, for instance, provide interpretable structures that reveal key factors influencing water quality, while Random Forest introduces randomness to handle complex data relationships effectively.

Support Vector Machines (SVM) excel in finding optimal hyperplanes that separate different classes, making them valuable for high-dimensional data analysis. Logistic Regression, on the other hand, models the probability of an instance belonging to a particular class, making it suitable for binary classification tasks like water potability assessment.

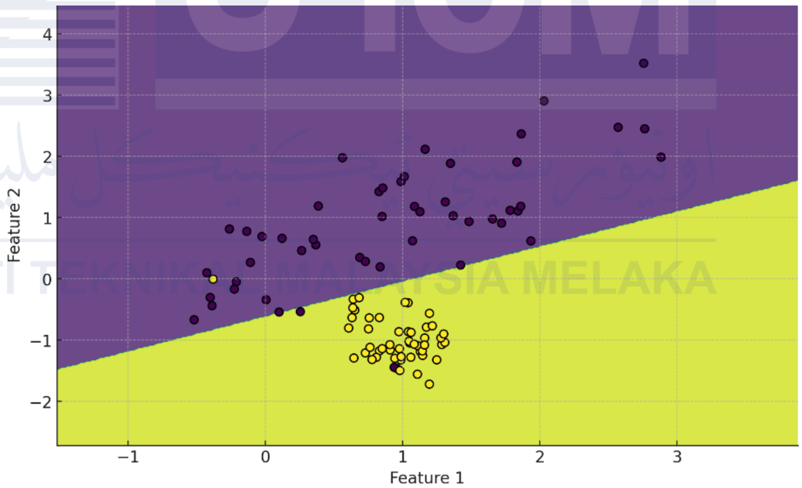

Figure 2.7 Support Vector Machine

The project aims to evaluate the performance of these ML models in detecting potable water based on sensor readings. Leveraging the strengths of various ML algorithms will enhance the accuracy and efficiency of water quality monitoring systems, ultimately contributing to the protection of valuable water resources.

2.3 Project Methodology

The development of the Water Quality Monitoring System using k-Nearest Neighbors (kNN) algorithm will follow a structured methodology to ensure systematic progress and effective implementation. The methodology encompasses the following key stages:

Stage 1: Data Collection

Data collection is a critical step in developing a machine learning model. A comprehensive dataset of water quality parameters will be acquired from reliable sources, including sensor readings for attributes such as temperature, pH levels, turbidity, and dissolved oxygen. The ESP32 microcontroller will be used to interface with temperature, Total Dissolved Solids (TDS), and pH sensors to collect real-time data. This data will be transmitted to a cloud database, such as Firebase, ensuring a continuous and up-to-date dataset. The dataset will include enough instances representing both potable and non-potable water samples, allowing the model to learn from diverse scenarios.

Stage 2: Data Preprocessing

In this stage, the collected data will undergo preprocessing to ensure its quality and suitability for training the model. Data cleaning techniques will be applied to handle missing values, outliers, and duplicates. The K-Nearest Neighbor (KNN) imputer will be used to handle any missing values in the dataset. Feature engineering methods will be employed to extract relevant features from the dataset that contribute most to water potability prediction. Additionally, strategies to address any imbalance in the dataset will be implemented to prevent bias towards the majority class.

Stage 3: Model Training and Testing

The pre-processed dataset will be divided into training and testing sets. The KNN algorithm, chosen for its simplicity and effectiveness in classification tasks, will be trained on the training set. Hyperparameter tuning may be performed to optimize the model's performance. The trained model will then be tested on the testing set to evaluate its accuracy and effectiveness in classifying water samples as potable or non-potable.

Stage 4: Model Evaluation

The performance of the trained KNN model will be evaluated using various metrics such as accuracy, precision, recall, and F1 score. This evaluation will provide insights into the model's ability to correctly classify water samples and identify any areas for improvement. Cross-validation techniques may be employed to ensure the robustness of the model.

Stage 5: Model Deployment

Upon satisfactory performance, the trained KNN model will be deployed in a real-world water quality monitoring system. The ESP32 will continuously collect sensor data (temperature, TDS, and pH levels) and transmit this data to Firebase. The KNN model will read the sensor readings from Firebase, classify the water samples in real-time, and predict whether the water is potable or non-potable. A user-friendly mobile application will be developed to enable users to interact with the system easily. The app will display the sensor readings and predictions, providing alerts or notifications when water quality parameters deviate from acceptable levels.

Stage 6: System Validation and Monitoring

Rigorous testing and validation procedures will be conducted to ensure the reliability, accuracy, and robustness of the deployed system. Continuous monitoring will be implemented to track the system's performance over time and address any issues or anomalies promptly. The ESP32 will play a crucial role in ensuring that the system remains functional and responsive by regularly sending updated sensor data to Firebase. By following this structured methodology, the Water Quality Monitoring System using the KNN algorithm aims to provide an effective and reliable solution for continuous monitoring and assessment of water quality parameters.

2.4 Project Requirements

2.4.1 Software Requirement

i. Integrated Development Environment (IDE):

- Visual Studio: The project will utilize Visual Studio as the primary integrated development environment for coding and software development tasks. Visual Studio offers robust features and support for various programming languages, facilitating efficient development workflows.

ii. Arduino IDE:

- Arduino IDE will be utilized for programming and uploading code to the Arduino microcontroller. It provides a user-friendly interface for writing, compiling, and uploading code to Arduino boards, essential for developing the hardware module of the Water Quality Monitoring System.

iii. Mobile Application Development:

- React Native: The project will leverage React Native, a popular JavaScript framework, for developing cross-platform mobile applications. React Native allows for efficient development of mobile apps for both iOS and Android platforms using a single codebase.
- Expo Go app: Expo Go will be used as a companion mobile app during the development phase. It enables quick testing and debugging of React Native applications on physical devices.

iv. Backend Services:

- Firebase: Firebase will serve as the backend platform for storing, managing, and synchronizing real-time data collected from the Water Quality Monitoring

System. It provides features such as database, authentication, and cloud functions, essential for building scalable and secure backend services.

v. Additional Tools:

- Text Editor: Visual Studio Code will be used alongside Visual Studio for writing code and scripts related to the project. Its lightweight and customizable nature makes it an ideal choice for software development tasks.
- Expo Go app: Expo Go app will be installed on physical devices for testing and debugging React Native applications directly on mobile devices.
- Draw.io: Draw.io will be used for creating system architecture diagrams and flowcharts, aiding in visualizing the system components and workflows efficiently.

2.4.2 Hardware Requirement

- Laptop Brand: Acer
- Model: Aspire 5
- Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
- Graphics Card: NVIDIA
- RAM: 8GB
- Operating System: Windows 10
- Arduino ESP32
- pH sensor
- Temperature sensor DS18B20
- Total dissolved solids (TDS) sensor
- Breadboard
- Jumper wires
- Power supply

2.4.3 Project Schedule and Milestones

The project schedule and milestones outline a structured timeline for developing the system. It includes various stages like planning, research, analysis, design, implementation, testing, and documentation spread over several weeks. Each phase focuses on key activities to ensure smooth progress, such as defining project requirements, designing system architecture, writing code, testing, and refining the system. By breaking the project into clear, manageable tasks, this schedule ensures that each aspect of the system is properly addressed, leading to the successful completion of the final product.

Table 2.1 Gantt Chart

Activity	**Week**																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Planning - Research - Project planning																						
Analysis - Analyse the project requirements. - Decide hardware and software necessary in building the system																						
Design - Design the system																						

Implementation - Create function program by produce lines of code.																						
- Implementation system																						
Testing - Test the system																						
Implementation - Continue working on coding for the incomplete modules in FYP 1																						
Testing - Test the overall functions of the system																						
Documentation - Presentation - System demonstration																						

2.4.4 Conclusion

Ultimately, the literature review can assist this project in identifying and describing the characteristics and needs associated with the proposed system. As a result, new research might be offered to improve the current system in which researchers can gain information by conducting research on other research papers released by the authors. Furthermore, in order to accomplish the project, the methodology must lead people in their comprehension of the domain scope and field. The following chapter will be devoted to analyzing the proposed system.

CHAPTER III: ANALYSIS

3.1 Introduction

This chapter will discuss the analysis of the proposed system for detecting potable water using k-Nearest Neighbors (kNN) based on sensor readings. The analysis ensures that the system meets the specified objectives and marks the start of the project lifecycle. The analysis will be separated into two parts: problem analysis and requirement analysis. The requirement analysis will be further subdivided into data requirements, functional requirements, non-functional requirements, and other necessary specifications. This comprehensive analysis will serve as the foundation for the design and development of the proposed system, ensuring its effectiveness in determining water potability.

3.2 Problem Analysis

The problem statement highlights the challenge of accurately detecting potable water using sensor data. To conduct an extensive problem analysis, we need to delve into the primary issues and components contributing to the problem. Here are three key aspects to consider:

- Sensor Data Variability: The accuracy and reliability of water quality sensors can vary due to environmental conditions, sensor aging, or calibration issues. Further analysis can be conducted to understand the extent of this variability and its impact on the performance of the k-Nearest Neighbors (kNN) algorithm. Techniques such as sensor fusion, data normalization, and regular sensor calibration can be explored to mitigate the effects of sensor variability and improve the reliability of water potability detection.

- False Negatives and Positives: An analysis can be conducted to assess the current rates of false negatives and false positives in the water potability detection system. False negatives (unsafe water classified as potable) pose a health risk, while false positives (safe water classified as non-potable) can lead to unnecessary interventions. Techniques such as adjusting classification thresholds, incorporating additional sensor data, or utilizing ensemble methods can be explored to minimize both false negatives and false positives, ensuring a high accuracy rate for detecting potable water.

- Dynamic Environmental Conditions: Environmental conditions such as temperature, pH, and contamination sources can change over time, affecting water quality. This entails reviewing historical water quality data for trends or patterns that may signal changes in water potability. Machine learning models can be updated and retrained with new data to detect and respond to these dynamic conditions effectively. Continuous system monitoring, regular data updates, and adaptive model retraining are required to ensure the system's ongoing effectiveness in detecting potable water under varying environmental conditions.

The problem revolves around the growing issue of water contamination due to urbanization and industrialization, which poses significant risks to both the environment and public health. Existing water quality monitoring techniques are often expensive, inaccessible, and unable to provide real-time data, limiting early detection and effective management of contamination. There is a pressing need for an affordable, accessible, and continuous monitoring solution, such as an Arduino-based system, that can provide real-time water quality assessments, offer a user-friendly interface, and enable rapid identification of environmental threats to ensure the safety of drinking water.

3.3 Requirement Analysis

3.3.1 Data Requirements

In our analysis of data requirements for the water quality assessment project, we begin by outlining the essential dataset that forms the core of our study. The dataset we are utilizing contains eleven key parameters that provide insights into the chemical and physical properties of water samples, crucial for predicting potability.

A	B	C	D	E	F	G	H	I	J	K
ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomet	Turbidity	Temperature	Potability
	204.8905	20791.32	7.300211873	368.5164	564.3086542	10.37978308	86.99097	2.963135	66.38504531	0
3.71608	129.4229	18630.06	6.635245884		592.8853591	15.18001312	56.32908	4.500656	32.18465486	0
8.099124	224.2363	19909.54	9.275883603		418.6062131	16.86863693	66.42009	3.055934	23.82530723	0
8.316766	214.3734	22018.42	8.059332377	356.8861	363.2665162	18.4365245	100.3417	4.628771	93.6061505	0
9.092223	181.1015	17978.99	6.546599974	310.1357	398.4108134	11.55827944	31.99799	4.075075	40.67717668	0
5.584087	188.3133	28748.69	7.544868789	326.6784	280.4679159	8.39973464	54.91786	2.559708	75.56350234	0
10.22386	248.0717	28749.72	7.513408466	393.6634	283.6516335	13.78969532	84.60356	2.672989	68.86787148	0
8.635849	203.3615	13672.09	4.563008686	303.3098	474.6076449	12.3638167	62.79831	4.401425	98.88006014	0
	118.9886	14285.58	7.804173553	268.6469	389.3755659	12.70604897	53.92885	3.595017	29.76460905	0
11.18028	227.2315	25484.51	9.077200017	404.0416	563.8854815	17.92780641	71.9766	4.370562	26.94852766	0
7.36064	165.5208	32452.61	7.550700907	326.6244	425.3834195	15.58681044	78.74002	3.662292	31.33040597	0
7.974522	218.6933	18767.66	8.110384501		364.0982305	14.5257457	76.48591	4.011718	27.90795082	0
7.119824	156.705	18730.81	3.606036091	282.3441	347.7150273	15.92953591	79.50078	3.445756	34.29762226	0
	150.1749	27331.36	6.838223471	299.4158	379.7618348	19.37080718	76.51	4.413974	86.99329471	0
7.496232	205.345	28388	5.072557774		444.6453523	13.2283111	70.30021	4.777382	96.07932125	0
6.347272	186.7329	41065.23	9.629596276	364.4877	516.7432819	11.53978119	75.07162	4.376348	68.31577096	0
7.051786	211.0494	30980.6	10.09479601		315.1412672	20.39702184	56.6516	4.268429	52.86398057	0
9.18156	273.8138	24041.33	6.904989726	398.3505	477.9746419	13.38734078	71.45736	4.503661	98.54555713	0
8.975464	279.3572	19460.4	6.204320859		431.44399	12.88875905	63.82124	2.436086	12.92130782	0
7.37105	214.4966	25630.32	4.43266929	335.7544	469.9145515	12.50916394	62.79728	2.560299	10.34417069	0
	227.435	22305.57	10.33391789		554.8200865	16.33169328	45.38282	4.133423	21.39918878	0
6.660212	168.2837	30944.36	5.858769131	310.9309	523.6712975	17.88423519	77.04232	3.749701	54.53494852	0

Figure 3.1 Dataset of various parameters of water quality

The dataset includes variables such as pH, hardness, solids concentration, chloramines level, sulfate content, conductivity, organic carbon concentration, trihalomethanes presence, turbidity, temperature, and the target variable, potability. These parameters are critical as they collectively define the quality of water and are instrumental in determining its suitability for consumption.

Unlike other projects where data may be sourced from multiple databases or APIs, our dataset is consolidated and sourced specifically for this project, ensuring consistency and relevance to our objectives.

Each record in the dataset represents a unique water sample with measured values across these parameters, providing a comprehensive view of the dataset's structure and content.

Our dataset does not currently exhibit significant issues such as missing data or imbalance among classes, as each parameter is well-represented across the dataset. However, as part of our data preprocessing phase, we will conduct detailed checks to ensure data quality, handle any outliers, and normalize feature distributions where necessary. These steps are crucial for preparing the dataset for subsequent machine learning model development and evaluation.

In summary, the data requirements for our project emphasize the importance of a well-curated dataset encompassing a comprehensive range of water quality parameters. Through careful preprocessing and utilization of advanced analytical techniques, we aim to harness the full potential of this data to build accurate predictive models for assessing water potability. This endeavor is pivotal for supporting informed decision-making in environmental monitoring and public health initiatives related to water quality assessment.

3.3.2 Functional Requirements

Functional requirements are essential to ensure the water quality assessment application fulfills its intended functionalities effectively and efficiently. The following functional requirements have been identified to guide the development and implementation phases:

- Data Collection and Integration: The system should collect real-time data from various sources, including sensors and manual inputs. It should integrate this data seamlessly into the existing dataset, ensuring a continuous flow of updated information for analysis.

- Data Preprocessing: Prior to analysis, the system must preprocess the collected data to ensure consistency and quality. This involves handling missing values, scaling numerical features, encoding categorical variables, and performing feature selection to optimize the dataset for modelling.

- Exploratory Data Analysis (EDA): The application should provide interactive visualizations and statistical summaries to facilitate exploratory data analysis. This includes histograms, scatter plots, correlation matrices, and summary statistics, enabling users to gain insights into the distribution and relationships among water quality parameters.

- Model Training and Evaluation: Utilizing machine learning algorithms, the system should train models to predict water potability based on historical data. It should employ techniques such as cross-validation to assess model performance and hyperparameter tuning to optimize model accuracy and generalizability.

- Prediction and Reporting: Once trained, the system should accept new input data and generate predictions on water potability in real-time. It should present results in a user-friendly interface, offering clear and actionable insights to stakeholders.

These functional requirements form the backbone of our water quality assessment system, ensuring robustness, reliability, and user-friendliness throughout the application lifecycle. By addressing these requirements systematically, we aim to develop a sophisticated tool that supports informed decision-making and enhances environmental monitoring efforts.

3.3.3 Non-functional Requirements

The non-functional requirements for the water quality assessment application ensure its performance, usability, security, and scalability. These requirements are crucial for ensuring the application's effectiveness and user satisfaction:

- Performance: The system should be capable of handling large volumes of water quality data efficiently. It must process data in real-time and provide timely responses to user queries and analyses. The response times for data retrieval, analysis, and visualization should be optimized to enhance user experience and decision-making.

- Accuracy: The analysis algorithms and models used by the system must exhibit high accuracy in predicting water potability. The system should minimize false positives and false negatives, ensuring that predictions are reliable and trustworthy for stakeholders. Continuous monitoring and evaluation of model performance are essential to maintain high accuracy over time.

- Security: Data security is paramount. The system should implement robust measures to protect sensitive water quality data from unauthorized access, breaches, or cyber-attacks. Encryption techniques, secure authentication mechanisms, and data anonymization should be employed to safeguard data integrity and confidentiality.

- Usability: The application should feature an intuitive and user-friendly interface designed for both technical users and stakeholders with varying levels of expertise in water quality analysis. It should include interactive visualizations, customizable dashboards, and easy-to-understand reports that facilitate data interpretation and decision-making.

- Scalability: As the volume and complexity of water quality data grow, the system should scale seamlessly to accommodate increased data processing and storage demands. It should support horizontal scalability by leveraging cloud infrastructure or distributed computing frameworks to ensure performance remains optimal under varying workloads.

These non-functional requirements underscore the importance of system reliability, security, usability, and scalability in delivering a robust water quality assessment application. By adhering to these requirements, we aim to create a system that not only meets but exceeds the expectations of its users and stakeholders, contributing to effective environmental monitoring and management.

3.3.4 Other Requirements

In addition to functional and non-functional requirements, the water quality assessment application has specific software and hardware requirements to support its development, deployment, and operation. These requirements ensure the application can effectively analyze and visualize water quality data.

Software Requirements

Table 3.1 outlines the essential software components necessary for developing and deploying the water quality assessment application:

Table 3.1 Software Requirements

Software Name	Description
Python 3.8+	Programming language used for implementing data analysis algorithms, machine learning models, and web application development.
VSCode	Interactive development environment for creating and sharing documents containing live code, equations, visualizations, and narrative text.

Pandas, NumPy	Python libraries for data manipulation and numerical computations, essential for preprocessing and analyzing water quality datasets.
Matplotlib, Seaborn	Python libraries for creating static, animated, and interactive visualizations to facilitate data exploration and presentation.
Scikit-learn	Implementation of the K-Nearest Neighbors algorithm in Scikit-learn for classifying water quality data based on the proximity to labeled training examples.
Firebase Realtime Database	Cloud-hosted NoSQL database to store and sync data in real-time, ensuring efficient data management and retrieval for the application.
Arduino IDE	Integrated Development Environment (IDE) for programming Arduino boards, enabling data acquisition and control for water quality sensors.
React Native	Framework for building native mobile applications using JavaScript and React, allowing the development of cross-platform mobile applications.
Expo Go	Toolchain for developing React Native applications, providing a fast and easy way to run React Native apps on iOS and Android devices.

Hardware Requirements

Table 3.2 specifies the hardware specifications necessary to support the development and deployment of the water quality assessment application:

Table 3.2 Hardware Requirements

Hardware Name	Description
Laptop Brand: Acer	The development environment will be hosted on an Acer Aspire 5 laptop, providing ample processing power and memory.
Processor	Intel Core i5 processor, capable of handling computational tasks efficiently.

RAM	8 GB RAM, ensuring sufficient memory for running data analysis algorithms and machine learning models effectively.
Operating System	64-bit Windows 10, providing a stable and compatible operating environment for software development and deployment.
Arduino Board	Arduino Uno or similar microcontroller board for collecting data from sensors and interfacing with the application.
Sensor Modules	Various water quality sensors, including pH, turbidity, conductivity, and temperature sensors, for real-time data monitoring.

These software and hardware requirements are essential for ensuring the water quality assessment application operates smoothly, facilitating robust data analysis, model training, and interactive visualization capabilities. By adhering to these requirements, the application can effectively support environmental monitoring efforts and decision-making processes related to water quality management.

3.4 Conclusion

In conclusion, the analysis phase has provided a robust framework for the upcoming design and implementation stages of the water quality assessment application. Through detailed problem and requirement analysis, we've identified critical challenges and outlined essential functionalities such as data validation, intuitive user interfaces, and accurate prediction using the K-Nearest Neighbors algorithm. These insights ensure the application meets performance, accuracy, and usability standards crucial for effective environmental monitoring and decision-making. The next phase will focus on detailed design, translating these requirements into a cohesive application architecture that aligns closely with project goals.

CHAPTER IV: DESIGN

4.1 Introduction

In this chapter, the design of the proposed technique for water quality assessment using AI techniques will be presented. The high-level design will include the system architecture, user interface design, and data handling strategies. The detailed design will delve into software components, hardware integration, and data processing methods. These design elements are critical for the successful implementation of the water quality assessment application, ensuring it meets the required performance, accuracy, and usability standards.

4.2 High-Level Design

The aim of this project is to develop a water quality assessment system using Artificial Intelligence techniques, which will be accessible through a web application. The AI techniques employed will analyze three key water quality parameters—pH, temperature, and solids—to predict the potability of water samples. This high-level design outlines the system architecture, which includes data collection, preprocessing, model training, and deployment phases. The system will evaluate water quality based on these parameters to determine the potability of the water. The effectiveness of the model will be evaluated by its accuracy in predicting water potability, ensuring reliable and actionable insights for users.

4.2.1 System Architecture for Expert System

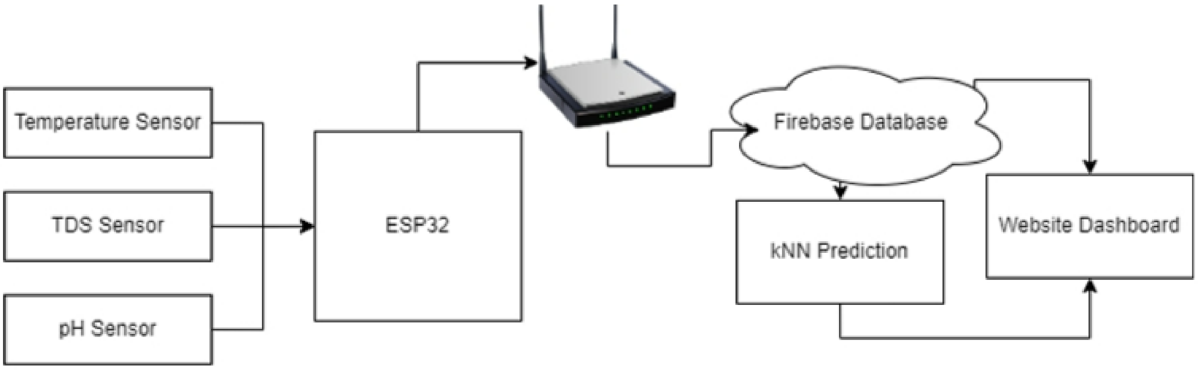

Figure 4.1 System Architecture

The water quality assessment system follows a well-defined system architecture that enables its functionality as an expert system. The architecture encompasses several

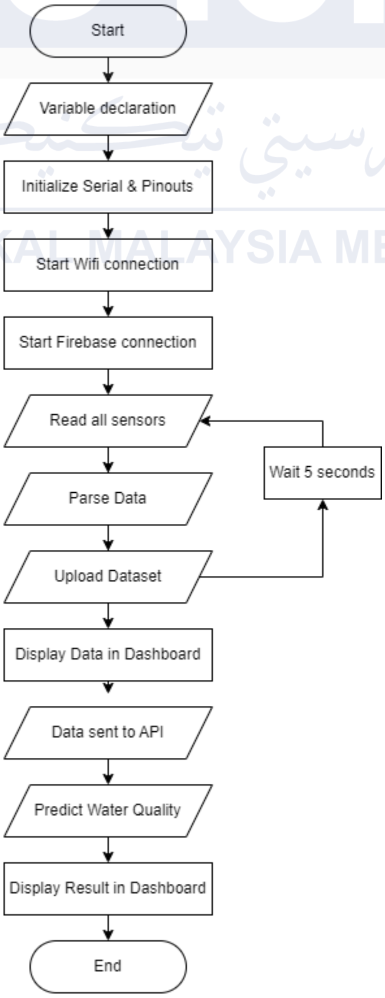

Figure 4.2 Flowchart of the Expert System

crucial components that work together seamlessly to analyze water quality based on pH, temperature, and solids parameters. Figure 4.1 illustrates the data flow of the machine learning model. The system architecture starts with the collection of water quality data, focusing on the parameters of pH, temperature, and solids. This data is then subjected to preprocessing, which includes procedures such as handling missing values and normalizing the data. Following preprocessing, the data is divided into training and testing sets to facilitate model development and evaluation.

The machine learning model, specifically a K-Nearest Neighbors (KNN) algorithm, is trained using the training data. This training process involves learning the patterns and relationships within the water quality parameters to accurately predict potability. The model's performance is then evaluated using the testing data to ensure its accuracy and reliability.

Once the model is trained, it is integrated into the system for real-time predictions. When a user inputs new water quality data through the web interface, the system processes this input, validates it, and feeds it into the trained KNN model. The model then evaluates the input parameters and predicts whether the water is potable or not. The prediction is subsequently displayed to the user through an intuitive interface, providing immediate and actionable feedback. To be precise, the system utilizes a dataset containing historical water quality measurements. This dataset undergoes preprocessing steps such as handling missing values and scaling features. The preprocessed data is then split into training and testing sets using a function such as train_test_split.

The core of the system is the KNN model, as illustrated in Figure 4.2. The model is trained on the preprocessed data, allowing it to learn the intricate patterns in the water quality parameters. When a user inputs new water quality data, the system processes this data and feeds it into the trained KNN model for prediction. The model evaluates the input parameters and classifies the water as either potable or non-potable. The prediction is then presented to the user through the web interface, providing immediate and clear feedback. Finally, the system is deployed as a web application using frameworks such as React Native and Expo Go for the front end, and Streamlit for backend integration. This deployment ensures that users can access the system easily and receive reliable water quality assessments based on the expert system's predictions.

4.2.2 User Interface Design for Expert System

The user interface design for the water quality assessment expert system is centered on providing users with a seamless and interactive experience. The system is launched on Expo Go, a framework for creating React Native applications, allowing for robust, cross-platform mobile app development. The primary goal of the user interface is to enable users to input water quality parameters and receive accurate predictions about

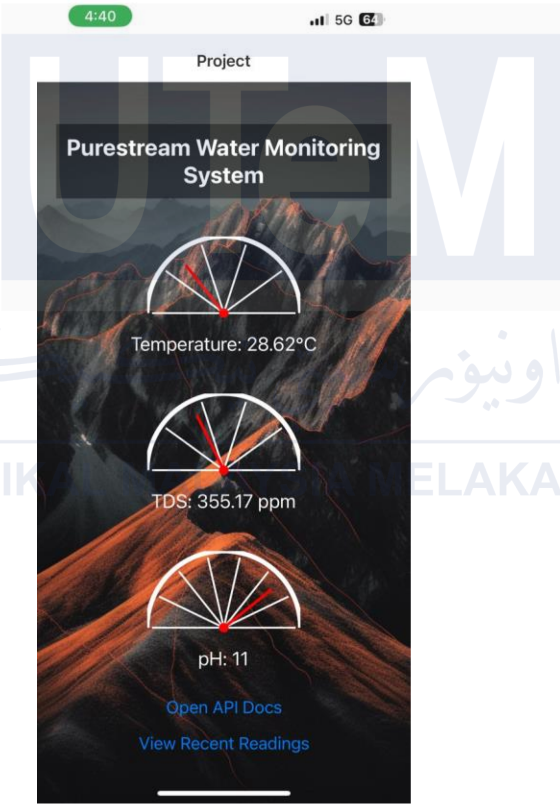

Figure 4.3 Main Input Screen for Water Quality Parameters

water potability. This design approach ensures a user-friendly and intuitive interaction with the system, thereby enhancing the overall user experience. The user interface is designed to be clean and straightforward, guiding users through the process of entering water quality data and receiving results. The main screen features input fields for the essential parameters: pH, temperature, and solids. Users can easily enter these values through touch-friendly controls. The layout is designed to be responsive, ensuring a smooth experience on both smartphones and tablets. After inputting the parameters, users can submit the data by tapping a clearly labeled "Predict" button. The app processes the input data and provides immediate feedback on the water's potability status. This feedback is displayed in a concise and informative manner, often accompanied by visual aids such as icons or color-coded indicators to enhance clarity.

Figure 4.4 Prediction Result Screen

Additionally, the app includes an information section where users can learn more about water quality parameters and their importance. This section is designed to educate users on how pH, temperature, and solids affect water potability, promoting a better understanding of the assessment process.

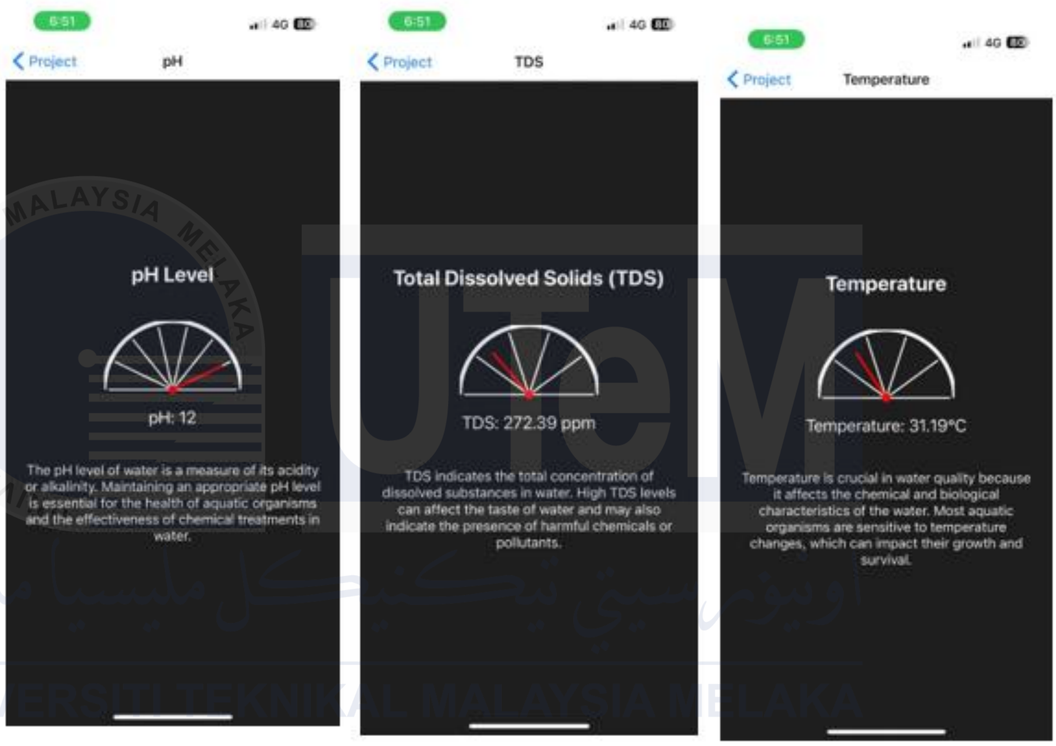

Figure 4.5 Information Section of each sensor's importance

The interface also integrates features to handle errors gracefully, such as notifications for missing or invalid inputs, ensuring that users can correct any mistakes before submission. This design philosophy ensures that the expert system is accessible and easy to use, even for individuals with limited technical expertise. In conclusion, the user interface design for the water quality assessment system is meticulously crafted to provide an intuitive and engaging user experience. By leveraging Expo Go and React Native, the application delivers a responsive and interactive platform that allows users to assess water quality accurately and effortlessly.

Navigation Design

The navigation design of the Expo Go mobile application for the water quality assessment expert system ensures a smooth and intuitive user experience. Figure 4.7 illustrates the flowchart navigation design of the app. This navigation design guides users seamlessly through the app, from launching it to receiving water quality predictions and accessing educational content. Users do not need to input values manually; instead, the values from the sensors are automatically displayed on the app.

- Main Monitoring Screen:
  Sensor Readings: The app displays real-time sensor readings for pH, temperature, and solids on interactive meters.
  Navigation: Users can view the current sensor readings and tap the "Analyze" button to proceed.

- Prediction Result Screen:
  Display Results: The app processes the sensor data and displays the water potability prediction, with clear indicators.
  Navigation: Users can tap "Back" to return to the Main Monitoring Screen.

- Information Section:
  Educational Content: This section provides detailed information about water quality parameters and their significance.
  Navigation: Users can return to the Prediction Result Screen or Main Monitoring Screen from here.

This navigation design ensures users can easily move through the application, from viewing real-time sensor data to receiving results and learning more about water quality. Each screen is designed to be intuitive, with clear instructions and seamless transitions, providing an efficient and user-friendly experience.

Input Design for Expert System

The input design for the water quality assessment expert system's mobile application captures sensor data for analysis. Users do not manually input values; instead, the application automatically retrieves and displays real-time sensor readings for pH, temperature, and solids. The user-friendly input interface, implemented using Expo Go, ensures that data is presented in an accessible and comprehensible manner. This design seamlessly integrates with the k-Nearest Neighbors (k-NN) model, which utilizes the sensor readings to generate accurate predictions about water quality.

- Real-time Updates: The app continuously retrieves data from the sensors monitoring pH, temperature, and solids in the water.
- Display Meters: The current values are displayed on interactive meters, providing a clear visual representation of the water quality parameters.
- Interactive Meters:
    - pH Meter: Shows the current pH level of the water, indicating acidity or alkalinity.
    - Temperature Meter: Displays the current water temperature.
    - Solids Meter: Indicates the concentration of solids in the water, which affects water clarity and quality.
- Integration with k-NN Model: The real-time sensor data is automatically fed into the k-NN model for analysis.
- Continuous Monitoring: The system continually monitors the water quality, updating the readings and predictions as new data is received.
- Instant Display: Users can see real-time changes in water quality parameters on their screens.
- Clear Indicators: The app uses color-coded indicators and icons to convey the status of each parameter (e.g., normal, warning, critical).

This input design ensures that users receive accurate, real-time information about water quality without the need for manual data entry. The automatic integration with the k-NN model allows for continuous monitoring and instant feedback, enhancing the overall user experience and reliability of the system.

Technical Design

The technical design of the water quality assessment expert system ensures compatibility and efficient data processing. It leverages the k-Nearest Neighbors (k-NN) algorithm, a robust classification technique, to evaluate water quality based on sensor readings of pH, temperature, and solids. The user interface is built using the Expo Go platform and React Native, providing a cross-platform compatible and user-friendly experience. Real-time sensor data is automatically displayed on the app's meters, offering immediate water quality assessments. The design includes interactive visualizations, such as meters for each parameter and color-coded status indicators, which enhance user understanding and interaction. Backend integration ensures that sensor data is collected and processed in real-time, with the k-NN model classifying water quality efficiently. The system prioritizes seamless integration, accurate predictions, and intuitive user interaction, maximizing its effectiveness in monitoring and assessing water quality.

Output Design

I'm unable to generate figures directly, but I can provide a detailed description of how figures and illustrations would be incorporated into the output design section for the water quality assessment expert system using Expo Go and React Native. The output design for the water quality assessment expert system is meticulously crafted to offer users a comprehensive and intuitive interface using Expo Go and React Native. This mobile application interface is specifically tailored to display real-time sensor data for pH, temperature, and solids parameters, providing stakeholders with immediate insights into water quality conditions.

- pH Parameter Display:

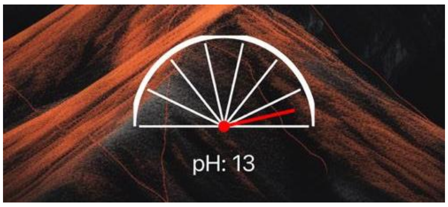

Figure 4.6 pH Parameter Display

The pH parameter displays in the mobile application. The pH meter is prominently featured on the interface, showing a color-coded range from acidic to alkaline. Users can visualize the current pH level of the water through a dynamic color gradient on the meter. This visual representation aids in quickly assessing the acidity or basicity of the water without needing to interpret numerical values alone.

- Temperature Parameter Display:

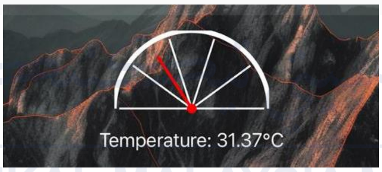

Figure 4.7 Temperature Parameter Display

The temperature parameter display using a thermometer-style meter within the mobile app. The meter visually indicates the current water temperature, providing users with an immediate understanding of whether the water temperature is within optimal ranges for various applications. Numerical temperature readings accompany the meter display, offering precise measurements for detailed analysis.

- Solids Parameter Display:

Figure 4.8 Solids Parameter Display

The solids parameter display in the mobile application interface. A gauge-style meter is utilized to depict the concentration of suspended solids or dissolved particles in the water. The gauge provides users with a visual representation of solids levels, facilitating quick assessments of water clarity and potential contamination. Numerical readings complement the gauge, ensuring accurate measurement and analysis of solids concentrations.

- Dynamic Data Updates:

The interface utilizes Expo Go's capabilities to dynamically update and display sensor readings in real-time. This ensures that users have access to the most current data, allowing them to monitor changes in pH, temperature, and solids levels as they occur. The real-time nature of the data presentation enhances the utility of the mobile app for both routine monitoring and rapid response to water quality events. The Database can be seen in the Recent Readings Page, for a certain time. The user can download the database in CSV if needed.

Date	Timestamp	pH	TDS	Temperature
28/08/2024	4:41:04 AM	11	0	28.69
28/08/2024	4:41:12 AM	11	0	28.69
28/08/2024	4:41:14 AM	11	0	28.62
28/08/2024	4:41:21 AM	11	333.42	28.62
28/08/2024	4:41:27 AM	11	334.3	28.62
28/08/2024	4:41:30 AM	11	334.59	28.62
28/08/2024	4:41:37 AM	11	334.88	28.62
28/08/2024	4:41:43 AM	11	109.5	28.56
28/08/2024	4:41:47 AM	11	0	28.62
28/08/2024	4:41:50 AM	11	0	28.56

Figure 4.9 Temporary In-App Database

- Enhanced Decision-Making:

The output design supports effective decision-making by presenting data in an intuitive and accessible manner. Visual indicators and numerical values empower stakeholders, such as environmental technicians, researchers, and policymakers, to interpret water quality data efficiently and respond promptly to changing conditions.

Finally, the output design of the water quality assessment expert system using Expo Go and React Native enhances user engagement and decision-making capabilities through intuitive visualizations, real-time data updates, and customizable alerts. By leveraging these technologies, the interface ensures accessibility, accuracy, and usability in monitoring and managing water resources effectively.

Conceptual and Logical Database Design

The conceptual and logical database design for the water quality assessment expert system focuses on efficient management and retrieval of sensor data within the mobile application environment. Each parameter such as pH, temperature, and solids is stored in separate data structures, ensuring streamlined access and manipulation. This approach supports real-time updates and historical tracking of water quality metrics, essential for monitoring changes and facilitating informed decision-making. By integrating Expo Go and React Native APIs, the system optimizes data retrieval while maintaining data integrity and security, ensuring that users receive accurate and timely information on water quality conditions directly through their mobile devices.

4.3 Detailed Design

For the detailed design of the water quality assessment expert system, several components and software tools are essential to ensure accurate data processing, analysis, and user interaction. At its core, the system utilizes a combination of Arduino IDE, Expo Go, React Native, and KNN algorithm for real-time data collection, processing, and prediction of water quality parameters including pH, temperature, and solids. Arduino IDE serves as the platform for programming the Arduino microcontroller to interface with water quality sensors. Sensors such as pH, temperature, and turbidity sensors are connected to the Arduino board, which continuously reads data from these sensors. Arduino IDE facilitates the coding of sensor data acquisition and communication protocols.

Expo Go, a mobile app development tool based on React Native, is used to build the mobile application interface. This interface allows users to view real-time water quality metrics directly on their mobile devices. Expo Go leverages React Native's capabilities for cross-platform development, ensuring compatibility across iOS and Android devices. React Native provides the framework for developing a responsive and user-friendly mobile application interface. It integrates seamlessly with Expo Go to create interactive visualizations and displays for pH, temperature, and solids data. React Native components such as sliders, graphs, and meters are utilized to present sensor readings in a clear and intuitive manner.

KNN algorithm (K-Nearest Neighbors) is employed for data prediction and anomaly detection based on historical sensor readings. It calculates the proximity of current sensor data points to previously recorded data points to determine the likelihood of water quality deviations or anomalies. This predictive capability enables proactive monitoring and timely intervention in case of water quality issues. To implement these components effectively, the system relies on specific software libraries and packages. This includes libraries for data visualization (e.g., matplotlib, seaborn) to display sensor data trends and anomalies graphically. Additionally, integration with Expo Go's APIs ensures seamless data communication between the mobile application and the Arduino microcontroller, facilitating real-time updates and user interaction.

By combining these tools and algorithms, the detailed design ensures robust data collection, processing, and visualization capabilities for monitoring water quality parameters. It enables users to access critical information promptly, empowering informed decision-making and proactive management of water resources.

4.3.1 Software or Hardware Design

The development of the water quality assessment system will adhere to an iterative and incremental approach, leveraging Agile methodology. Agile is chosen for its flexibility in accommodating changes and enhancements throughout the development lifecycle, which is crucial for adapting to evolving requirements and user feedback in real-time monitoring applications.

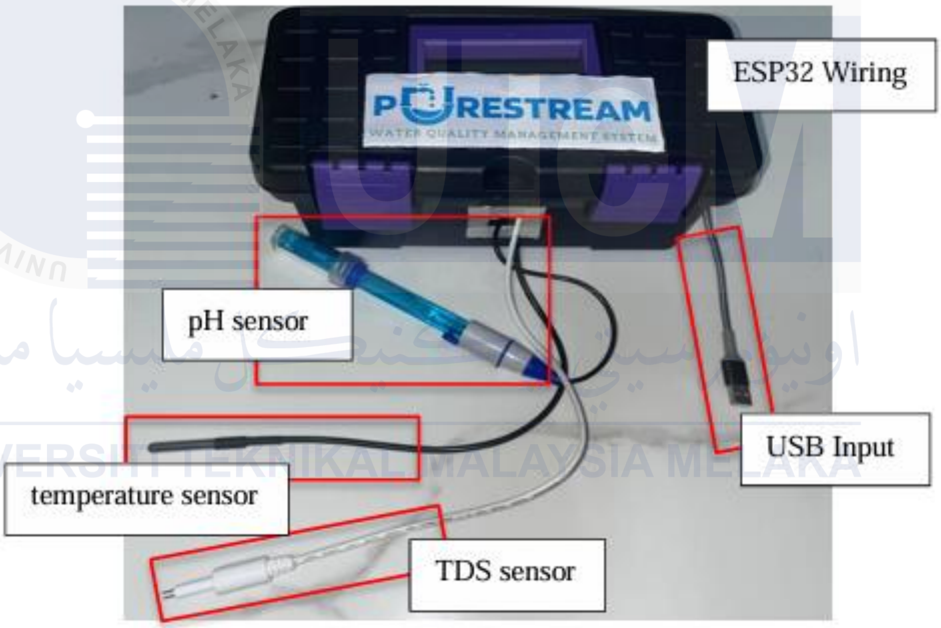

Figure 4.10 PureStream Product Prototype

The hardware design revolves around integrating Arduino microcontrollers with sensors to capture real-time data from water quality parameters such as pH, temperature, and solids. Each sensor is interfaced with the Arduino board, which processes the sensor readings and transmits them wirelessly using Bluetooth or Wi-Fi modules to the mobile application.

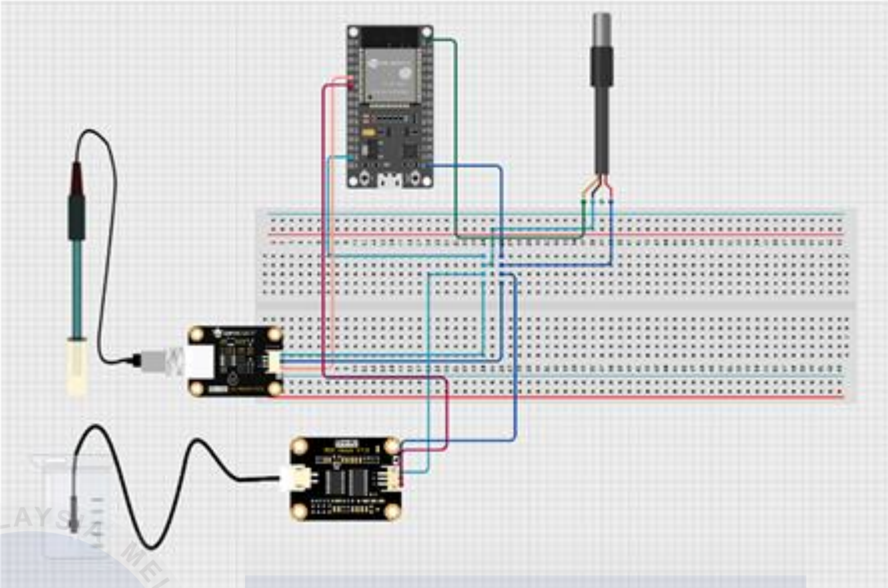

Figure 4.11 Schematic Circuit with Breadboard

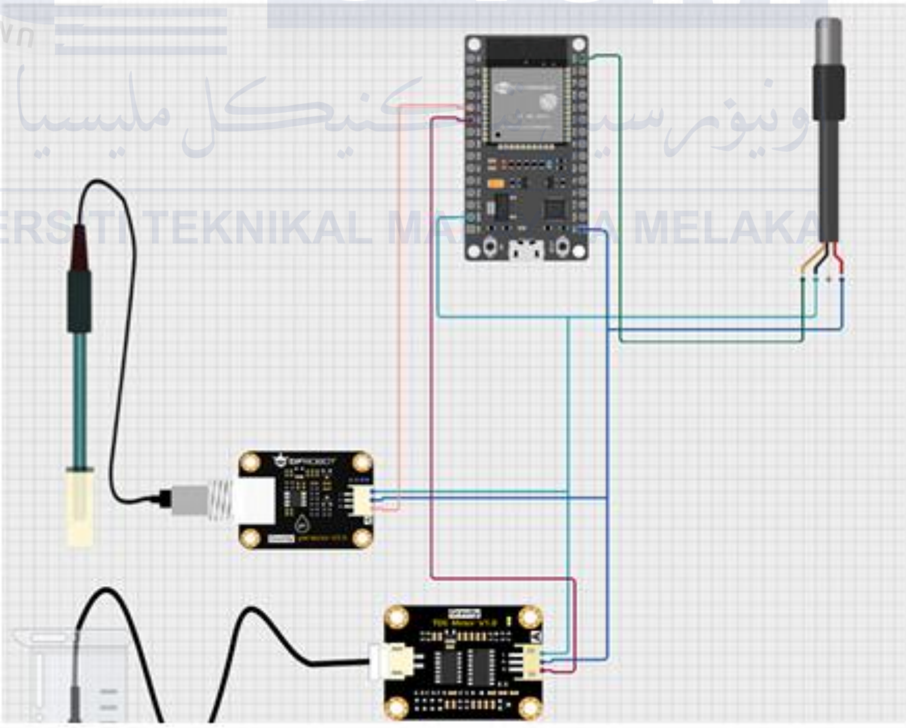

Figure 4.12 Schematic Circuit Net View

Table 4.1 Description of Waterfall Model

Phase	Description
System Analysis	During this phase, a comprehensive analysis of the water quality monitoring system requirements is conducted. This includes identifying the parameters to be monitored (pH, temperature, solids), data collection methods (sensors interfaced with Arduino), and user interface requirements (Expo Go and React Native for mobile application).
System Design	In this phase, the system architecture is designed to integrate Arduino microcontrollers with sensors and establish wireless communication for data transmission to the mobile application. Detailed design specifications include hardware selection, sensor calibration, and software interfaces (Expo Go for mobile UI design).
Implementation	The implementation phase involves coding and development activities. Arduino IDE is used to program microcontrollers for sensor data acquisition and transmission. Expo Go and React Native are employed to develop a user-friendly mobile interface that displays real-time water quality metrics.
Testing	The system is thoroughly tested to ensure functionality, reliability, and accuracy. Testing includes sensor calibration checks, data transmission validation, and mobile application interface responsiveness testing. Any issues identified are addressed before moving to deployment.
Result	Once testing is successfully completed, the water quality assessment system is deployed for real-

	world use. This phase involves setting up the system in the field, ensuring all components work together seamlessly, and providing user training on system operation and maintenance.

4.3.2 Physical Database Design

For the water quality assessment expert system, the system relies on real-time sensor data transmitted from Arduino microcontrollers to a mobile application interface. Therefore, there are no specific physical database design considerations for this project. The system architecture primarily focuses on data acquisition, transmission, and visualization through integrated hardware and software components rather than traditional database storage and management. This approach ensures that sensor data, such as pH levels, temperature readings, and solids content, are captured and processed in real-time without the need for persistent storage in a database. The emphasis is on immediate data availability and visualization for users through Expo Go and React Native interfaces, enhancing the system's responsiveness and usability for water quality monitoring and assessment purposes.

4.4 Conclusion

In this chapter, the design of the water quality assessment expert system has been thoroughly explored and articulated. The design process encompassed various critical aspects to establish a solid framework for system implementation. The high-level design provided an overview of the system architecture, user interface design, and navigation flow. This architectural blueprint ensures seamless integration between hardware components like Arduino microcontrollers and software platforms such as Expo Go and React Native for mobile application interfaces, optimizing real-time data acquisition and visualization.

Furthermore, the detailed design focused on technical aspects such as the integration of sensor data (pH, temperature, solids) into the Expo Go and React Native interfaces, ensuring accurate and real-time display on graphical meters. This design approach emphasizes usability and accessibility, enabling stakeholders to monitor water quality parameters effectively. With the completion of the design phase, we are wellpositioned to move forward to the implementation stage. The design choices made during this chapter will guide the development process, ensuring the successful realization of an efficient water quality assessment system that supports informed decision-making and enhances environmental monitoring practices.

CHAPTER V: IMPLEMENTATION

5.1 Introduction

This chapter details the implementation of the water quality monitoring system, including the setup of development environments, tool configurations, and the system's status. Firebase Realtime Database was used to store sensor data from Arduino IDE and ESP32 microcontrollers. FastAPI was employed to develop a backend service for water quality predictions using a K-Nearest Neighbors (KNN) model. The frontend, built with React Native, provides real-time data visualization and user interaction. Visual Studio Code was the primary IDE for the entire development process. The implementation process focused on ensuring a seamless flow of data from sensors to the cloud and then to the user interface, providing real-time insights and predictions about water quality.

5.2 Software Development Environment setup

As shown in Figure 5.1, Arduino IDE was used as the primary tool for developing and testing the microcontroller code, specifically for the ESP32 board and connected sensors. The Arduino IDE provides a user-friendly platform that allows seamless code writing, uploading, and debugging for the hardware components of the water quality monitoring system. This setup is crucial for ensuring the accurate and reliable operation of the sensors used in the project.

Figure 5.1 Arduino IDE

In parallel, Visual Studio Code (VS Code), was selected as the integrated development environment (IDE) for developing both the backend and frontend of the application. Python, a versatile and widely used programming language, was the core language for implementing the FastAPI backend. FastAPI, as shown in Figure 5.3, was chosen for its high performance and ease of use in building RESTful APIs, which are essential for processing sensor data and providing real-time water quality predictions.

The backend logic, including data handling, prediction algorithms, and API endpoints, was developed in Python within VS Code. Firebase was integrated as the real-time database for storing sensor readings and syncing them across devices. The combination of Firebase with FastAPI allows for efficient storage, retrieval, and processing of sensor data, ensuring that the water quality predictions are based on the most recent and accurate readings.

For the frontend, React Native was employed, along with Expo Go streamlines the testing and deployment process on mobile devices, enabling real-time updates and debugging. This environment was essential for developing a cross-platform mobile application that presents sensor data and predictions in an intuitive and user-friendly manner. This well-structured development environment, combining Arduino IDE, Visual Studio Code, Python, FastAPI, Firebase, and React Native, enabled a cohesive and efficient workflow. Each tool was chosen for its compatibility with the project's specific needs, contributing to the successful implementation and deployment of the water quality monitoring system.

5.3 Software Configuration Management

5.3.3 Configuration Environment Setup

In this section, the configuration environment setup for the water quality monitoring system will be discussed. The focus is on establishing a controlled and efficient environment where all components of the software development process align harmoniously.

A key component of this setup is the use of the Arduino IDE, a powerful tool for developing and managing the firmware for the ESP32 microcontroller and connected sensors. The Arduino IDE was downloaded from the official Arduino website, and version 1.8.19 was installed and launched. The Arduino IDE allows us to write, upload, and debug the firmware, ensuring that the ESP32 and the connected water quality sensors operate correctly.

Next, Visual Studio Code (VS Code), version 1.81.1, was utilized as the primary IDE for developing the backend and frontend of the system. The environment was set up to support the FastAPI framework, which is used to build the RESTful API for processing sensor data and generating water quality predictions. Python 3.9 was chosen as the programming language, and the necessary packages, including FastAPI, Pydantic, and Uvicorn, were installed and managed within this environment.

To ensure real-time data synchronization and storage, Firebase Realtime Database was integrated into the system. The Firebase environment was configured through the Firebase Console, where a new project was created to store and manage sensor data. The Firebase SDK was then integrated into both the backend and the React Native frontend, ensuring seamless communication between the components.

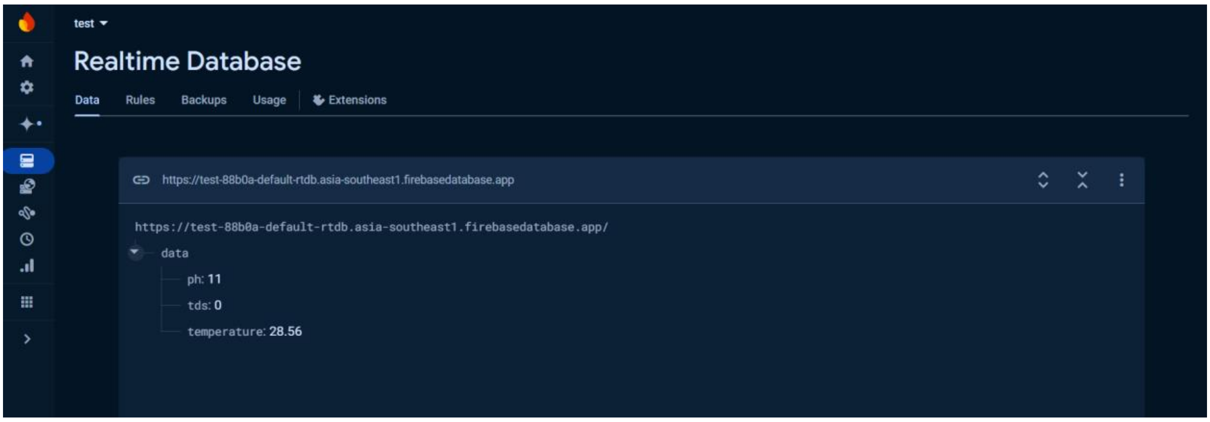

Figure 5.2 Firebase Console

The React Native frontend, developed using Expo Go, required careful configuration to ensure compatibility with the backend and Firebase. The Expo CLI was installed and used to create and manage the React Native project. The development environment was meticulously set up to handle the cross-platform requirements of the mobile application, which displays real-time sensor readings and water quality predictions to users.

Finally, the entire development environment, including the Arduino IDE, VS Code, Firebase, and Expo Go, was configured to ensure a streamlined workflow. This setup enables efficient development, debugging, and deployment of the water quality monitoring system, ensuring that all components work together harmoniously.

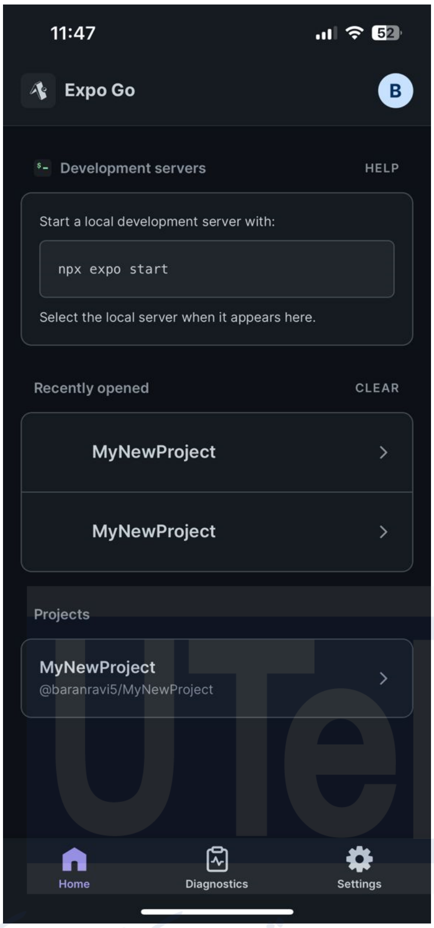

Figure 5.3 Expo Go Interface

This configuration environment setup is essential for maintaining the project's integrity, allowing for smooth transitions between development stages, and ensuring that the water quality monitoring system operates reliably and effectively.

5.3.2 Version Control Procedure

The version control procedure for the water quality monitoring system is a crucial aspect of maintaining consistency and reliability throughout the development process. Table 5.1 provides a detailed list of the key packages and tools installed within the project. These packages are essential for ensuring the smooth operation of the system, from data collection and processing to real-time monitoring and prediction.

This practice not only ensures consistency across different development environments but also simplifies the process of identifying and resolving potential issues that may arise during the project's lifecycle. By keeping track of the specific versions of each package, the development team can avoid compatibility issues and ensure that all components of the system function as expected.

Table 5.1 List of Packages Installed

Package/Tool Name	Version	Package/Tool Name	Version
Arduino IDE	2.1.0	Firebase SDK	11.9.0
Python	3.9.13	React Native	0.72.3
FastAPI	0.85.0	Expo Go	49.0.3
Uvicorn	0.18.2	Axios	1.4.0
Pydantic	1.9.1	dotenv	16.0.3

5.4 Implementation Status

Within this section, a comprehensive overview of the implementation status for various facets of this project were provided. To effectively monitor and convey the project's development, the implementation status are segmented into two distinct categories which are "Development Implementation Status" and "Module Analysis Implementation Status". These categories are thoughtfully presented and organized in Table 5.2 and 5.3, respectively, allowing stakeholders to gain clear insights into the project progression.

Table 5.2 Development Implementation Status

Task	Duration Completed (days)	Date Completed
Research and Data Collection	35	30 April 2024
Proposed Technique	20	20 May 2024
Literature Review	21	10 June 2024
Design Model	15	25 June 2024
Others	15	10 July 2024

Table 5.3 Module Analysis Implementation Status

Module Name	Description	Duration Completed (days)	Date Completed
Hardware Module	Calibration and integration of sensors (DS18B20, TDS sensor, pH sensor) with Arduino and ESP32	12	15 June 2024
Artificial Intelligence Module	Development of FastAPI backend for water quality prediction, including model training and integration.	14	29 June 2024
System Module	Implementation of Firebase for real-time data storage, React Native with Expo Go for frontend, and system integration.	18	17 July 2024

5.5 Conclusion

To conclude, the implementation phase has been important in the project. Here, the ideas and plans were converted into tangible software solutions to ensure that the system can successfully implement, as certain software and packages only work on specified version, and that it knows what to do for the next process. The next chapter will be discussing the testing phase of the project.

CHAPTER VI: TESTING

6.1 Introduction

In this chapter, testing is conducted to validate the functionality, performance, and reliability of the water quality monitoring system. This phase is crucial for ensuring that the system operates as expected, free from bugs, and is ready for deployment. Through systematic testing, we aim to identify and address any potential issues, ensuring the system meets the objectives.

6.2 Test Plan

The test plan is a detailed document outlining the test strategy, objectives, schedule, resources, and timelines necessary for the successful execution of the water quality monitoring system project.. By consolidating all relevant information into this document, we ensure that it can be easily reviewed by management and potentially reused in future projects.

6.2.1 Test Organization

The testing phase of the water quality monitoring system involves several key individuals, each assigned specific roles and responsibilities critical to the success of the testing process. In this section, the organization of the testing phase is outlined, and Table 6.1 details the roles and corresponding responsibilities of those involved. These individuals play a vital role in thoroughly examining the system, identifying any issues, and ensuring that the system meets all project requirements.

Table 6.1 Roles and responsibilities involved in the testing phase

Roles	Responsibility
Developer	The person who is responsible for overall project scheduling. The person must check all the project's process and testing result.
Course Mate	Acts as a user to test the system, providing feedback on usability and functionality.
Supervisor	The person as a Supervisor of Final Year Project to use and test the system.

6.2.2 Test Environment

Testing of the system is carried out in a working environment. There are two types of setups in this system.

Table 6.2 Test Environment

Component	Tool	Description
Hardware	Operating System	Microsoft Windows 11, 64-bit
	Processor	Intel(R) Core(TM) i5 CPU @ 2.90GHz
	Memory/RAM	16 GB
	Storage or Hard Disk	Solid State Drive (SSD) – 240 GB
	Input Devices	Keyboard and Mouse
	Arduino ESP 32	Microcontroller
	Arduino Sensors	pH, Temperature and TDS Sensors
Software	Arduino IDE	Used for programming and calibrating the sensors connected to the Arduino board.
	Visual Studio Code	The primary code editor for developing the system's backend and frontend components, including FastAPI and React Native.
	Firebase	Stores and retrieves real-time sensor data during testing and development.

	FastAPI	Handles API requests, processes sensor data, and returns water quality predictions.
	React Native with Expo Go	Used to develop and test the mobile application that interfaces with the system.
	Browser	Used to access and test the system's web interface, including API documentation and real-time data display.

6.2.3 Test Schedule

Table 6.3 Test Schedule

Testing Task	Description	Duration(days)	Remarks
Unit Testing	To test individual components of the water quality monitoring system to ensure they function as expected.	3	OK
Sensor Integration Testing	To verify that all sensors (temperature, TDS, and pH) are properly integrated and delivering accurate data to Firebase.	3	OK
API Testing	To test the FastAPI backend, ensuring it correctly processes sensor data and returns accurate predictions.	2	OK
Mobile App Interface Testing	To test the React Native mobile application, ensuring it correctly displays real-time sensor data and predictions.	2	OK
Data Export Testing	To test the functionality of exporting sensor data and	1	OK

	predictions to CSV/Excel files via the app.		
System Integration Testing	To ensure that the entire system, including hardware, backend, and frontend, operates cohesively.	4	OK

6.3 Test Strategy

In the testing phase, two primary strategies are employed: white box testing and black box testing.

White box testing is a method used to examine the internal structure, logic, and code of the software application. For this project, white box testing was applied to the data processing and prediction tasks. This approach required a deep understanding of the system's code to ensure its quality and correctness. Given its time-consuming nature, white box testing was conducted during the system's development to detect and fix bugs early. The focus was on verifying the programming code, including the integration of sensor data, API functionality, and the accuracy of the water quality predictions.

Black box testing focuses on assessing the functionality and performance of the user interface without requiring knowledge of the system's internal workings. This method was applied after the system's development to ensure it functioned as intended from the user's perspective. The black box testing evaluated how well the mobile application handled inputs like sensor data and user interactions, ensuring the system reliably displayed real-time data and predictions.

6.3.1 Classes of tests

The testing process involved different classes of tests, as shown in Table 6.4.

Table 6.4 Classes of tests

Classes of Test	Functional Testing	i. White box testing ii. Black box testing
	Non-functional Testing	**i. Usability Testing** Conducted to evaluate the user experience and interface of the mobile app. This testing assessed how intuitive and user-friendly the app was, identifying any issues that could affect user interaction. **ii. Graphical User Interface (GUI) Testing** Involved a detailed examination of the app's graphical elements, such as input fields, text alignment, and image placement, to ensure that the visual layout met design standards and enhanced the user experience.

6.4 Test Implementation

6.4.1 Experimental/ Test Description

In this section, we identify and outline all the test cases to ensure that the water quality monitoring system meets its requirements and functions correctly. Each test case includes steps, expected results, descriptions, and status to evaluate system performance. The goal is to verify the accuracy of sensor data processing, prediction results, and user interactions. Tables 6.5 and 6.6 provide a detailed description of the test cases involved.

Table 6.5 Model Classification Testing Description

Test Case Number	Action	Expected Result	Actual Result
T001	Input dataset.	Sensor readings (temperature, TDS, pH) are recorded and uploaded to Firebase.	OK
T002	Pre-processing.	Sensor data is cleaned, formatted, and split into training and testing sets.	OK
T003	Train the machine learning model.	Train models (e.g., XGBoost, Random Forest) on water quality data to compare performance.	OK
T004	Machine learning model evaluation.	Display and compare performance metrics (e.g., accuracy) of different models.	OK
T005	Prediction process.	System predicts water quality (potable or not potable) based on sensor data.	OK

Table 6.6 Credit Card Fraud Detection Testing Description

Test Case Number	Action	Expected Result	Actual Result
T011	Run the mobile app and backend services.	The water quality monitoring app displays real-time sensor data and predictions.	OK
T012	Enter sensor readings manually (if applicable).	Entered data is processed and displayed correctly in the app.	OK
T013	Export data to CSV.	Recent sensor readings and predictions are exported and saved in CSV format.	OK
T014	Verify data accuracy and prediction results.	Data displayed matches the sensor readings and predictions made by the model.	OK

6.4.2 Test Data

In the context of analyzing the water quality dataset, the data was meticulously divided into distinct subsets to facilitate the training, validation, and testing of machine learning models. This approach ensures that the models are effectively trained and evaluated to accurately predict water quality.

The dataset was first split into training and testing subsets. 60% of the dataset was allocated for training purposes. This training dataset, denoted as X_train for sensor readings (features) and Y_train for water quality labels (potable or not potable), was utilized to instruct the machine learning models on how to recognize patterns related to water quality.

The remaining 40% of the data was further divided into two equal parts:

Validation Dataset: 20% of the original dataset was designated as X_val and Y_val. This validation dataset played a crucial role in assessing the model's performance and fine-tuning hyperparameters during training.

Testing Dataset: The remaining 20% was designated as X_test and Y_test. This testing dataset was reserved for the final evaluation phase. It was employed to rigorously assess the trained machine learning models' ability to generalize to new, unseen water quality data.

In simple terms, 60% of the data was used to train the models on water quality detection. The remaining 40% was split into two equal parts: one part was used to validate the models' performance during training, and the other part was used to test the models' ability to predict water quality accurately on new data that the models had not seen before.

6.5 Test Result and Analysis

System: Water Quality Monitoring System

Module/Unit: Data Collection and Machine Learning Model

Table 6.7 Test Analysis for Data Collection and Machine Learning Model

Testing Number	Action	Expectation	Results	Remarks
T001	Input dataset.	The system should successfully record sensor readings (temperature, TDS,	PASS	Sensor readings are successfully stored in Firebase.

		pH) and upload them to Firebase Realtime Database without any data loss or errors.		Checked data consistency.
T002	Pre-processing.	Sensor data should be cleaned (e.g., removing noise, handling missing values), correctly formatted, and split into training and testing sets without errors.	PASS	Data was cleaned and split as expected. Verified by checking data structure.
T003	Train the machine learning model.	Models (e.g., XGBoost, Random Forest) should be trained on water quality data, and their performance metrics (e.g., accuracy) should be calculated and logged.	PASS	Training completed successfully for all models. Performance metrics recorded.
T004	Machine learning model evaluation.	The system should display and compare performance metrics (e.g., accuracy, precision, recall) of different models, with clear and correct outputs.	PASS	Comparison of models displayed correctly. Verified metrics and model selection.

T005	Prediction process.	The system should accurately predict water quality (potable or not potable) based on sensor data inputs, with predictions stored and displayed appropriately.	PASS	Predictions are accurate and match expected outcomes. Validated with test data.

System: Water Quality Monitoring System

Module/Unit: Mobile Application and Data Export

Table 6.8 Test Analysis for Mobile Application and Data Export

Testing Number	Action	Expectation	Results	Comments/Notes
T011	Run the mobile app and backend services.	The water quality monitoring app should display real-time sensor data and predictions without delays or errors in the interface.	PASS	Real-time data display is smooth and accurate. No lag or errors observed.
T012	Enter sensor readings manually (if applicable).	Manually entered sensor data should be processed and displayed correctly in the	PASS	Manual input works as expected. Data displayed accurately.

		app, matching the expected format and accuracy.		
T013	Export data to CSV.	The system should export recent sensor readings and predictions to a CSV file without any data loss, with correct formatting and content.	PASS	CSV export was successful. Verified file content and structure.
T014	Verify data accuracy and prediction results.	Data displayed in the app should match the sensor readings and predictions made by the model, ensuring no discrepancies or errors.	PASS	Data accuracy confirmed. Predictions and sensor readings are consistent.

6.6 Conclusion

In conclusion, the testing phase is crucial for verifying that all components of the system function correctly and deliver accurate results. This chapter outlines the various tests conducted to ensure optimal performance. The next chapter will focus on evaluating the project's overall effectiveness, discussing its strengths and weaknesses, and offering recommendations for improvement.

CHAPTER VII: CONCLUSION

7.1 Observation on Weaknesses and Strengths

This chapter discusses the strengths and weaknesses observed throughout the Water Quality Monitoring System project. The system's integration of real-time sensor data and predictive analytics using machine learning models is a significant strength, providing users with immediate feedback on water quality. The user interface is intuitive, allowing users to easily monitor temperature, TDS, and pH levels, and to view predictions on water potability. Additionally, the ability to export data to CSV enhances the system's usability for analysis and reporting purposes.

However, the system has certain limitations. While it effectively processes and displays sensor data, the accuracy of the predictions could be improved further, particularly in edge cases where sensor readings are near critical thresholds. Moreover, the system currently lacks a feature for continuous model learning based on new data inputs, which could enhance prediction accuracy over time. The dependency on internet connectivity for data storage and retrieval through Firebase may also pose a limitation in remote areas with poor connectivity.

7.2 Propositions for Improvements

To address the identified weaknesses, several improvements are proposed. Fine-tuning the machine learning models, particularly the K-Nearest Neighbors (KNN) model, could help reduce false positives and negatives. Implementing a dynamic data validation process to handle sensor anomalies more effectively would also enhance the system's reliability. Adding a feature for continuous model retraining based on incoming sensor data could improve the accuracy of predictions over time.

Additionally, enabling offline data storage and syncing with Firebase when the internet is available could make the system more robust in remote areas.

7.3 Project Contribution

This project contributes to the field of environmental monitoring by providing an accessible and user-friendly Water Quality Monitoring System that leverages IoT and machine learning technologies. By offering real-time insights into water quality, it aids in ensuring safe water consumption and highlights the potential of integrating predictive analytics into environmental monitoring systems. The system's design also serves as a foundation for further research and development in real-time monitoring applications.

7.4 Conclusion

In conclusion, this project has successfully identified and addressed the key challenges in real-time water quality monitoring. While it showcases significant strengths in providing timely and accurate predictions, there is still room for improvement, particularly in enhancing model accuracy and system resilience. The integration of IoT and advanced machine learning techniques offers scalability and adaptability, enabling the system to be deployed in diverse environments and continuously updated with new data. With future enhancements, such as improved sensor technologies, more robust algorithms, and expanded data integration, this system holds great promise for broader applications in ensuring water safety, preventing contamination, and ultimately improving public health outcomes.

REFERENCES

Mohiddin, M., Bodapally, K., Siramdas, S., Shainaz, N., & Sriramula, S. K. (2019). A real time low cost water quality progress recording system using Arduino Uno board. In *Learning and analytics in intelligent systems* (pp. 201–209). https://doi.org/10.1007/978-3-030-24318-0_24

Gokulanathan, S., Manivasagam, P., Prabu, N., & Venkatesh, T. (2019). GSM based water quality monitoring system using Arduino. Shanlax International Journal of Arts Science and Humanities, 6(4), 22–26. https://doi.org/10.34293/sijash.v6i4.341

Hong, W., Shamsuddin, N., Abas, E., Apong, R., Masri, Z., Suhaimi, H., Gödeke, S., & Noh, M. (2021). Water Quality Monitoring with Arduino Based Sensors. Environments, 8(1), 6. https://doi.org/10.3390/environments8010006

Araneta, A. a. S. (2022). Design of an Arduino-Based water quality monitoring system. International Journal of Computer Science and Mobile Computing, 11(3), 152–165. https://doi.org/10.47760/ijcsmc.2022.v11i03.017

Faustine, A., Mvuma, A. N., Mongi, H. J., Gabriel, M. C., Tenge, A. J., & Kucel, S. B. (2014). Wireless Sensor Networks for Water Quality Monitoring and Control within Lake Victoria Basin: Prototype Development. Wireless Sensor Network, 06(12), 281–290. https://doi.org/10.4236/wsn.2014.612027

Aghenta, L. O., & Iqbal, M. T. (2019). Low-Cost, open source IoT-Based SCADA system design using Thinger.IO and ESP32 thing. Electronics, 8(8), 822. https://doi.org/10.3390/electronics8080822

Padma, P. S., Priya, A. V. N. S. B. K. S., Sree, D. P., & Pavani, G. (2023). IOT based water quality Monitoring system. International Journal for Research in Applied Science and Engineering Technology, 11(10), 1975–1979. https://doi.org/10.22214/ijraset.2023.56357

Rao, T. S., Pranay, P., Narayana, S., Reddy, Y., Sunil, N., & Kaur, P. (2021). ESP32 based implementation of water quality and quantity Regulating system. Atlantis Highlights in Computer Sciences/Atlantis Highlights in Computer Sciences. https://doi.org/10.2991/ahis.k.210913.016

Moroney, L. (2017). The Firebase Realtime database. In Apress eBooks (pp. 51–71). https://doi.org/10.1007/978-1-4842-2943-9_3

Hameed, M. S., Ramya, E., Pillai, L., Kirubavathy, K., Shylesh, D. D., Bai, V. T., & Jaganathan, R. (2022). Realtime monitoring system for surface water quantity and analyzing of water quality using microcontroller. Journal of Pharmaceutical Negative Results, 13(SO3). https://doi.org/10.47750/pnr.2022.13.s03.073

Lathkar, M. (2023). High-Performance Web Apps with FastAPI. In Apress eBooks. https://doi.org/10.1007/978-1-4842-9178-8

Bansal, P., & Ouda, A. (2022). Study on integration of FastAPI and machine learning for continuous authentication of behavioral biometrics. 2022 International Symposium on Networks, Computers and Communications (ISNCC). https://doi.org/10.1109/isncc55209.2022.9851790

Králusz, T. A. (2024). Mobile Application Development with React Native and Leveraging Third-Party Libraries.

Lakshmikantha, V., Hiriyannagowda, A., Manjunath, A., Patted, A., Basavaiah, J., & Anthony, A. A. (2021). IoT based smart water quality monitoring system. Global Transitions Proceedings, 2(2), 181-186.

Moparthi, N. R., Mukesh, C., & Sagar, P. V. (2018, February). Water quality monitoring system using IoT. In 2018 Fourth international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB) (pp. 1-5). IEEE.

Das, B., & Jain, P. C. (2017, July). Real-time water quality monitoring system using Internet of Things. In 2017 International conference on computer, communications and electronics (Comptelix) (pp. 78-82). IEEE.

APPENDIX A

Sample of Dataset

ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Temperature	Potability
	204.8904555	20791.31898	7.300211873	368.5164413	564.3086542	10.37978308	86.99097046	2.963135381	66.38504531	0
3.716080075	129.4229205	18630.05786	6.635245884		592.8853591	15.18001312	56.32907628	4.500656275	32.18465486	0
8.099124189	224.2362594	19909.54173	9.275883603		418.6062131	16.86863693	66.42009251	3.05593375	23.82530723	0
8.316765884	214.3733941	22018.41744	8.059332377	356.8861356	363.2665162	18.4365245	100.3416744	4.628770537	93.6061505	0
9.092223456	181.1015092	17978.98634	6.546599974	310.1357375	398.4108134	11.55827944	31.99799273	4.075075425	40.67717668	0
5.584086638	188.3133238	28748.68774	7.544868789	326.6783629	280.4679159	8.39973464	54.91786184	2.559708228	75.56350234	0
10.22386216	248.0717353	28749.71654	7.513408466	393.6633955	283.6516335	13.78969532	84.60355617	2.672988737	68.86787148	0
8.635848719	203.3615226	13672.09176	4.563008686	303.3097712	474.6076449	12.3638167	62.79830896	4.401424715	98.88006014	0
	118.9885791	14285.58385	7.804173553	268.6469407	389.3755659	12.70604897	53.92884577	3.595017181	29.76460905	0
11.18028447	227.2314692	25484.50849	9.077200017	404.0416347	563.8854815	17.92780641	71.97660103	4.370561937	26.94852766	0
7.360640106	165.5207973	32452.61441	7.550700907	326.6243535	425.3834195	15.58681044	78.74001566	3.662291783	31.33040597	0
7.974521649	218.6933005	18767.65668	8.110384501		364.0982305	14.5257457	76.48591118	4.011718108	27.90795082	0
7.119824384	156.7049933	18730.81365	3.606036091	282.3440505	347.7150273	15.92953591	79.50077834	3.445756223	34.29762226	0
	150.1749234	27331.36196	6.838223471	299.4157813	379.7618348	19.37080718	76.50999553	4.413974183	86.99329471	0
7.496232208	205.3449822	28388.00489	5.072557774		444.6453523	13.2283111	70.30021265	4.777382337	96.07932125	0
6.347271761	186.7328807	41065.23476	9.629596276	364.4876872	516.7432819	11.53978119	75.07161729	4.376348291	68.31577096	0
7.0517858	211.0494061	30980.60079	10.09479601		315.1412672	20.39702184	56.65160379	4.268428858	52.86398057	0
9.181560007	273.8138067	24041.32628	6.904989726	398.3505168	477.9746419	13.38734078	71.45736221	4.503660796	98.54555713	0
8.975464348	279.3571668	19460.39813	6.204320859		431.44399	12.88875905	63.8212371	2.43608559	12.92130782	0
7.371050302	214.4966105	25630.32004	4.43266929	335.7544386	469.9145515	12.50916394	62.79727715	2.560299148	10.34417069	0
	227.4350484	22305.56741	10.33391789		554.8200865	16.33169328	45.38281518	4.133422644	21.39918878	0
6.660212026	168.2837469	30944.36359	5.858769131	310.9308583	523.6712975	17.88423519	77.04231805	3.749701241	54.53494852	0
	215.9778587	17107.22423	5.607060453	326.9439777	436.256194	14.18906221	59.85547583	5.459250956	99.13881449	0
3.902475686	196.9032467	21167.5001	6.996311586		444.4788825	16.60903316	90.18167588	4.528522696	45.84375135	0