

**Tribhuvan University**

**Orchid International College**

**A FINAL YEAR PROJECT REPORT**

**On**

**WATER QUALITY CLASSIFICATION USING RANDOM FOREST ALGORITHM**

**Under the supervision of**

**Er. Dhiraj Kumar Jha**

**Orchid International College**

**Submitted To**

**Department of Computer Science and Information Technology**

**Orchid International College**

**In partial fulfillment of the requirement for the Bachelor Degree in Computer**

**Science and Information Technology**

**Submitted by**

**Nischal Wagle (23878/076)**

**Sital Gurung (23891/076)**

**Sneha Tuladhar (23892/076)**

**March, 2024**



(Affiliated to Tribhuvan University)

**SUPERVISOR’S RECOMMENDATION**

I hereby recommend that the report prepared under my supervision by Nischal Wagle (23878/076), Sital Gurung (23891/076) and Sneha Tuladhar (23892/076) entitled **“WATER QUALITY CLASSIFICATION USING RANDOM FOREST ALGORITHM”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

**………………………….**

**Er. Dhiraj Kumar Jha**

Project Coordinator

Department of BSc. CSIT

Orchid International College

Bijayachowk, Gaushal



(Affiliated to Tribhuvan University)

**LETTER OF APPROVAL**

This is to certify that this project prepared by Nischal Wagle (23878/076), Sital Gurung (23891/076) and Sneha Tuladhar (23892/076) entitled “Water Quality Classification Using Random Forest Algorithm” in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **…………………………**  **Er. Dhiraj Kumar Jha**  Project Supervisor,  Department of Computer Science and IT Orchid International College  Bijaychowk, Gaushala | **………………………**  **Er. Dhiraj Kumar Jha**    Internal Examiner,  Department of Computer Science and IT Orchid International College  Bijaychowk, Gaushala |
| **……………………….. External Examiner**  Central Department of Computer Science and IT Tribhuvan University  Kirtipur, Nepal | |

# ACKNOWLEDGEMENTS

It is truly a privilege to express our profound gratitude to all those who supported us. We want to express our appreciation to **Er. Dhiraj Kumar Jha** sir and **Shikha Sharma** ma’am for their invaluable assistance in completing this report. Their expert guidance and willingness to dedicate time from their busy schedules for document reviews have been instrumental. The continuous inspiration we received from them played a pivotal role in achieving our goals.

Additionally, we would like to extend our sincere thanks to the faculty of Orchid International College, Department of Computer Science and Information Technology, for providing us with this opportunity. Our gratitude also goes to our friends and colleagues for their selfless efforts in aiding us to complete the project.

**Nischal Wagle (23878/076)**

**Sital Gurung (23891/076)**

**Sneha Tuladhar (23892/076)**

# ABSTRACT

Water Quality Classification emerges as a pivotal application of Machine Learning, involving a sophisticated learning process aimed at forecasting water quality levels. This system operates by collecting user-provided details and predicting water quality levels in string format, employing the Random Forest Algorithm for accurate classification. The application is constructed as a Model-View-Controller (MVC) web system using the Flask Framework, where HTML facilitates page design, Bootstrap CSS enhances styling, JavaScript for the minimal functioning of the loaded pages and Python serves as the backend framework, incorporating the Random Forest Algorithm as the primary predictive model. The implementation of this system offers a robust and efficient tool for accessing water quality, contributing to advancements in sustainable water resource management.

**Keywords: *Water Quality Classification, Machine Learning, Random Forest, HTML/Bootstrap CSS, JavaScript, Flask, Python.***

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENTS i](#_Toc160917275)

[ABSTRACT ii](#_Toc160917276)

[LIST OF FIGURES v](#_Toc160917277)

[LIST OF TABLES vi](#_Toc160917278)

[CHAPTER 1- INTRODUCTION 1](#_Toc160917279)

[1.1 Introduction 1](#_Toc160917280)

[1.2 Problem Statement 1](#_Toc160917281)

[1.3 Objectives 1](#_Toc160917282)

[1.4 Scope and Limitation 2](#_Toc160917283)

[1.5 Development Methodology 2](#_Toc160917284)

[1.6 Report Organization 3](#_Toc160917285)

[CHAPTER 2- BACKGROUND STUDY AND LITERATURE REVIEW 4](#_Toc160917286)

[2.1 Background Study 4](#_Toc160917287)

[2.2 Literature Review 5](#_Toc160917288)

[CHAPTER 3- SYSTEM ANALYSIS 6](#_Toc160917289)

[3.1 System Analysis 6](#_Toc160917290)

[3.1.1 Requirement Analysis 6](#_Toc160917291)

[3.1.2 Feasibility study 11](#_Toc160917292)

[3.1.3 Process Analysis 12](#_Toc160917293)

[CHAPTER 4 - SYSTEM DESIGN 14](#_Toc160917294)

[4.1 Design 14](#_Toc160917295)

[4.1.1 Class Diagram 14](#_Toc160917296)

[4.1.2 Activity Diagram 15](#_Toc160917297)

[4.1.3 Sequence Diagram 16](#_Toc160917298)

[4.2 Algorithm Details 17](#_Toc160917299)

[CHAPTER 5- IMPLEMENTATION AND TESTING 19](#_Toc160917300)

[5.1 Implementation 19](#_Toc160917301)

[5.1.1 Analysis and Design Tools 19](#_Toc160917302)

[5.1.2 Implementation Tools (Frontend and Backend) 19](#_Toc160917303)

[5.1.3 Description of the Algorithm 20](#_Toc160917304)

[5.2 Testing 25](#_Toc160917305)

[5.2.1 Test Plan 25](#_Toc160917306)

[5.2.2 Unit Test Cases 26](#_Toc160917307)

[CHAPTER 6- CONCLUSION AND ENHANCEMENTS 30](#_Toc160917308)

[6.1 Limitations of Project 30](#_Toc160917309)

[6.2 Conclusions 30](#_Toc160917310)

[6.3 Future Enhancements 30](#_Toc160917311)

[REFERENCES 31](#_Toc160917312)

[APPENDICES 32](#_Toc160917313)

# LIST OF FIGURES

[[Figure 1.1 Software Development Life Cycle………………………………....................... 3](#_heading=h.2grqrue)](#_Toc160663662)

[[Figure 3.1 Use Case Diagram 7](#_heading=h.2grqrue)](#_Toc160663663)

[[Figure 3.2 Gantt Chart 12](#_heading=h.2grqrue)](#_Toc160663664)

[[Figure 3.3 ER- Diagram 13](#_heading=h.2grqrue)](#_Toc160663665)

[[Figure 4.1 Class Diagram 14](#_heading=h.2grqrue)](#_Toc160663666)

[[Figure 4.2 Activity Diagram 15](#_heading=h.2grqrue)](#_Toc160663667)

[[Figure 4.3 Sequence Diagram 16](#_heading=h.2grqrue)](#_Toc160663668)

[[Figure 4.4 Random Forest Algorithm 17](#_heading=h.2grqrue)](#_Toc160663669)

[[Figure 5.1 Result using Random Forest 24](#_heading=h.2grqrue)](#_Toc160663670)

[[Figure 5.2 Result Using Decision Tree 25](#_heading=h.2grqrue)](#_Toc160663671)

# LIST OF TABLES

[[Table 3.1 Functional Requirements 6](#_heading=h.1egqt2p)](#_Toc160661965)

[[Table 3.2 Use Case Description for User Registration/Login 8](#_heading=h.1egqt2p)](#_Toc160661966)

[[Table 3.3 Use Case Description for browse Water Quality Prediction 8](#_heading=h.1egqt2p)](#_Toc160661967)

[[Table 3.4 Use Case Description for Inserting Parameters 9](#_heading=h.1egqt2p)](#_Toc160661968)

[[Table 3.5 Use Case Description for Parameter Validation 10](#_heading=h.1egqt2p)](#_Toc160661969)

[[Table 3.6 Use Case Description for Viewing Prediction 10](#_heading=h.1egqt2p)](#_Toc160661970)

[[Table 5.1 Model Accuracy 24](#_heading=h.1egqt2p)](#_Toc160661971)

[[Table 5.2 Test Case for Register 26](#_heading=h.1egqt2p)](#_Toc160661972)

[[Table 5.3 Test Case for Login Page 27](#_heading=h.1egqt2p)](#_Toc160661973)

[[Table 5.4 Test Case for User Prediction 28](#_heading=h.1egqt2p)](#_Toc160661974)

[[Table 5.5 System Testing 29](#_heading=h.1egqt2p)](#_Toc160661975)

# CHAPTER 1- INTRODUCTION

## **1.1 Introduction**

The Water Quality Prediction System is a web-based application that helps users to decide and determine water quality. The Water Quality Prediction System leverages the Random Forest algorithm to enhance the assessment and management of water resources. This web-based application utilizes a predictive modeling approach, where the algorithm processes user inputs encompassing critical water quality parameters such as Faecal content, Oxygen levels, pH, Sediment, Temperature, Nitrogen, Phosphorus, and Turbidity. The Random Forest algorithm, known for its ensemble learning capabilities, makes predictions based on collective insights from diverse data sources. This website aims to help the community as a utility to help decide and determine water quality. This feature is expected to help in better management of water resources.

## **1.2 Problem Statement**

Water pollution is a big problem for communities, making it hard to keep water safe. Right now, there's no good way to quickly predict how clean the water is. This makes it tough to manage and protect water resources. Contamination and deterioration of water quality pose significant challenges to communities. The lack of a robust system for real-time water quality prediction hinders the efficient management of water resources. This system seeks to address this gap by implementing a predictive model to assess water quality based on various parameters.

## **1.3 Objectives**

**1.3.1 General Objectives**

* Develop a predictive model using the Random Forest Algorithm for water quality assessment.
* Create a user-friendly web-based platform for inputting water quality parameters.
* Map and identify water quality areas based on user input.
* Provide real-time predictions to assist in water resource management.
* Predict diseases by determining water quality.

**1.3.2 Specific Objectives**

* Design and implement Random Forest Algorithm for water quality prediction.
* Develop an intuitive user interface for data input on the web-based platform.
* Integrate GIS functionalities for mapping and identifying water quality areas.
* Analyzed water quality parameters correlated with disease outbreaks.
* Classify water quality assessments into general and specific objectives for reporting and analysis.

## **1.4 Scope and Limitation**

The scopes of the project are:

* Prediction of water quality using Fecal, Oxygen, pH, Sediment, Temperature, Nitrogen, Phosphorus, and Turbidity parameters.
* User input in the range of 0 to 100 for each parameter.
* Integration of machine learning model with a web interface using the Flask framework.

The limitations of the project are:

* The model's accuracy is dependent on the quality and quantity of training data.
* The model assumes linear relationships between input parameters and water quality.
* External factors not considered in the model may influence water quality.

## **1.5 Development Methodology**

In the context of software development, a methodology serves as the guiding framework that structures, plans, and oversees the intricate process of crafting an information system. Throughout the Software Development Life Cycle, various models are employed to navigate this journey. In the creation of our Water Quality Prediction System, we have opted for the prototype model, given the system's confined functional requirements and the stability of user specifications. The prototype methodology is an iterative system development model wherein a prototype is constructed, tested, and iteratively refined until an acceptable and efficient prototype is attained. Often referred to as a dummy implementation, this model involves designing a preliminary system with defined parameters. Based on the system's output, iterative refinements or rework are undertaken. In our system's context, an initial prototype is constructed with specific parameters, subjected to testing, and, if it meets predetermined accuracy standards, embraced as the final version. If not, a cycle of refinement persists until an acceptable level of accuracy is realized.

.

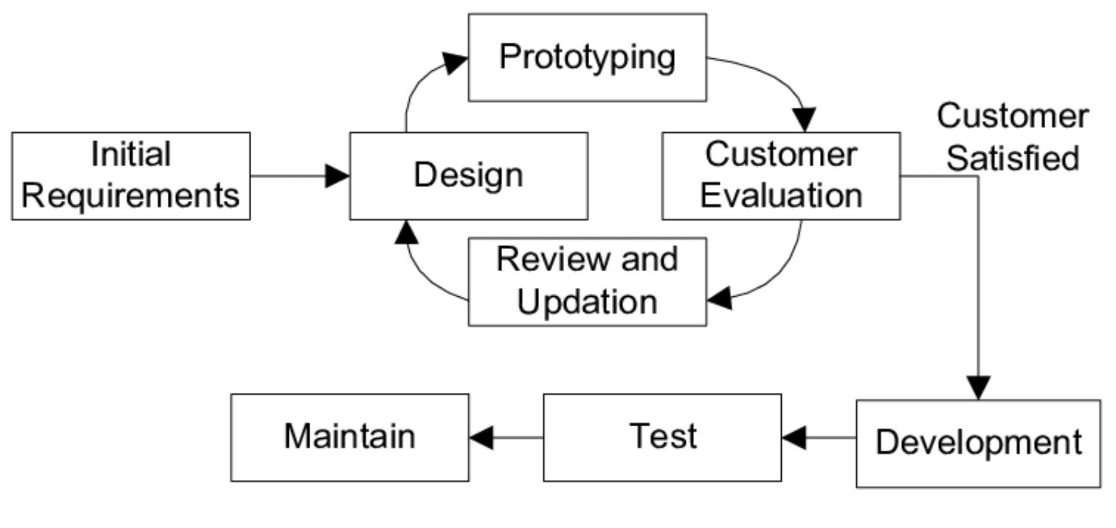


Figure 1.1 Software Development Life Cycle

## **1.6 Report Organization**

The report is organized as follows:

* Chapter 1: Introduction
* Chapter 2: Literature Review
* Chapter 3: Methodology
* Chapter 4: Implementation
* Chapter 5: Results and Discussion
* Chapter 6: Conclusion and Future Work

# CHAPTER 2- BACKGROUND STUDY AND LITERATURE REVIEW

## **2.1 Background Study**

Monitoring water quality is crucial for our health, environment, and even industries. While traditional methods exist, they're often slow, expensive, and limited in scope. Luckily, new technologies are emerging like web apps and machine learning algorithms that can predict water quality faster, cheaper, and across wider areas. The Water Quality Prediction System is one such example, using user-input data and advanced algorithms to map and predict water quality, helping us manage this precious resource more effectively.

A water quality prediction system is a web-based application that aims to help users decide and determine water quality. This feature is expected to help in better management of water resources. This system has a feature that can help people identify water quality based on the content in the water which consists of Fecal, Oxygen, pH, Sediment, Temperature, Nitrogen, Phosphorus, and Turbidity. The model used is the Random Forest Classifier ensemble tree-based learning algorithm. This will return the result of an array of model predictions from all inputs that have been entered. Output predictive labels in the form of obtaining real legible results. Then, this integer value is converted to a result string based on the label that matches the sign. The result of this process is series of predictive outcomes, such as "Excellent", "Good", "Fair", "Marginal", and "Poor".

Existing water quality prediction systems often need help in meeting diverse and personalized user requirements. One prominent limitation lies in the uniformity of predictions, as some systems generate similar outcomes for different users without considering the distinct preferences and environmental contexts. Additionally, sparse data issues, particularly in historical check-in data, may impede the accuracy of predictions, limiting the system's effectiveness. Furthermore, achieving real-time predictions and adapting to dynamic changes in water quality conditions poses another significant challenge for these systems. Overcoming these limitations and incorporating improvements in personalization, handling sparse data, and enhancing real-time prediction capabilities are essential steps toward advancing the efficacy of water quality prediction systems.

## **2.2 Literature Review**

In paper [1] Alomani1, Shahd Maadi, and Aaeshah Alhakamy propose a novel approach using PySpark and the Random Forest algorithm to predict river water potability based on 10 key features: pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, turbidity, and potability status.Model achieved outstanding results, predicting water potability with 86 % accuracy and an F1-score of 1.0. This signifies the model's potential for real-world applications in water quality monitoring.

In paper [2] Bachir Sakaa ,Samir Boudibi ,Ahmed Elbeltagi ,Hicham Chaffai proposed a study that explored two AI models to predict river water quality in the Wadi Saf-Saf basin. Surprisingly, the simpler Random Forest model outperformed a hybrid approach, even with just five key parameters like pH and temperature. This suggests a cost-effective, real-time solution for water quality monitoring. However, larger datasets and testing on diverse rivers are needed to fully unlock the potential of AI for sustainable river management.

In paper [3] Gakii and Jepkoech approach a study that employs a Classification model utilising a decision tree to analyse water quality data from various counties in Kenya. The goal is to predict whether water is clean or not based on parameters like Alkalinity, pH level, and conductivity. The application of decision tree as a data mining method facilitates the prediction of clean water, streamlining the work of laboratory technologists by identifying which samples should undergo further analysis. The study utilised secondary data from the Kenya Water Institute and implemented the model in WEKA software. Five decision tree classifiers were employed, with J48 achieving the highest accuracy at 94%, while Decision Stump had the lowest accuracy at 83%

# CHAPTER 3- SYSTEM ANALYSIS

## **3.1 System Analysis**

System analysis is the process of defining the elements of a system such as the architecture, modules and the components, the different interfaces of those components and the data that goes through that system. In this project, we have used Draw.io design tools.

### **3.1.1 Requirement Analysis**

Requirement analysis holds the process of reviewing and determining the system need, functional requirements and non-functional requirements that a system must meet. For requirement analysis, following approaches are followed:

**Functional Requirements**

The functional requirement for a system describes what the system should do. Those requirements depend on the type of software being developed, the expected users of the software. The system should provide the statement of services, how the system should react to particular inputs and how the system should behave in particular situations.

Table 3.1 Functional Requirements

|  |  |
| --- | --- |
| Identifier | Requirements |
| User Registration | Users can register for an account by providing a unique username and password. |
| User Login | Registered users can securely log in to the system using their username and password. |
| Browse Water Quality Information | Authenticated users can easily browse water quality information on the internet. |
| Insert Parameters | Authenticated users can input specific parameters for prediction. |
| Parameter Validation | The system validates user-inputted parameters to ensure accuracy and prevent errors. |

**Use Case Diagram**

The following Use Case Diagram describes the action of the system and how the interaction happen between user and admin

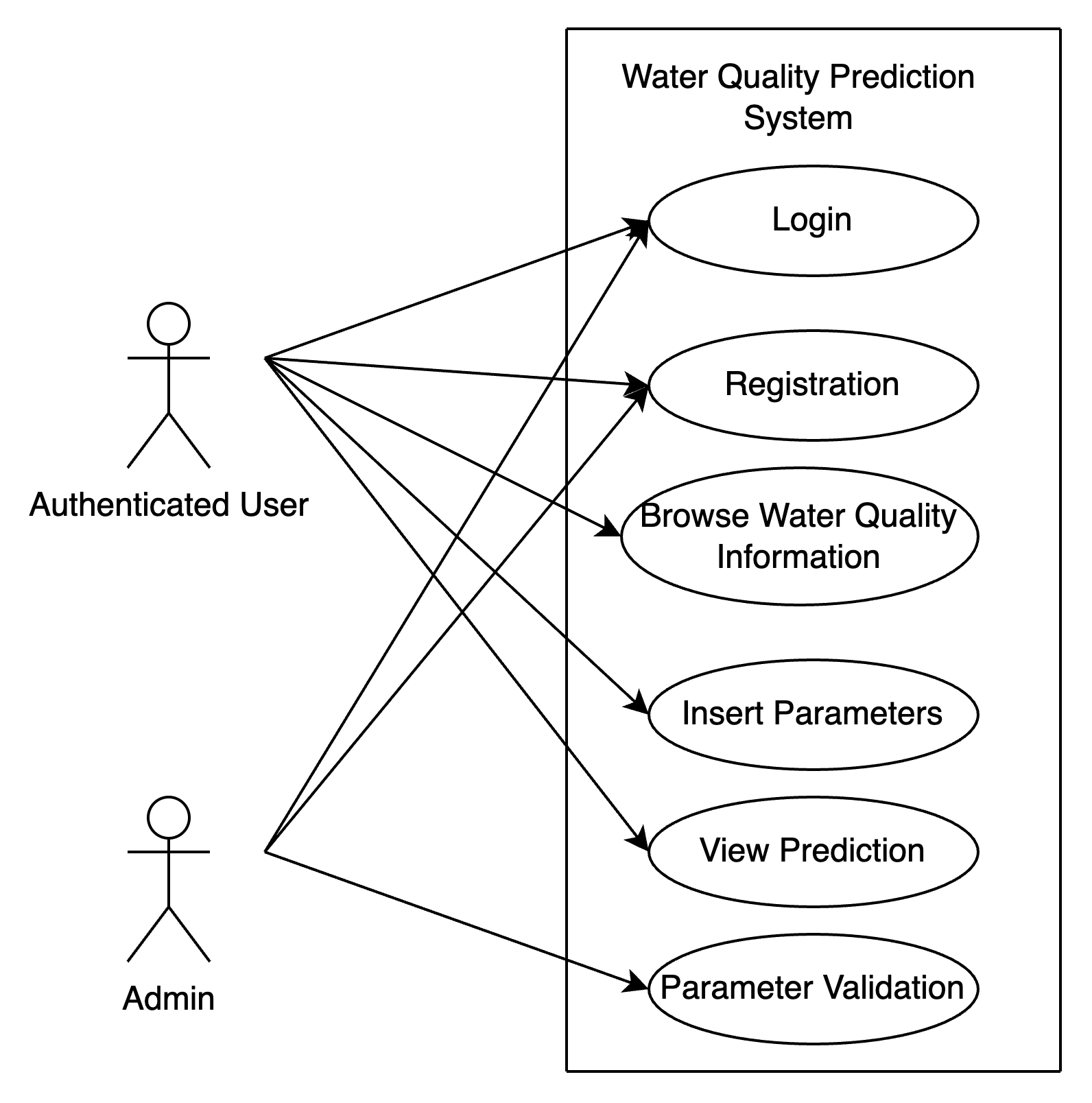


Figure 3.1 Use Case Diagram

**Use Case Description**

Table 3.2 Use Case Description for User Registration/Login

|  |  |
| --- | --- |
| Case Name | Registration/Login |
| Primary Actor | Authenticated user, admin |
| Secondary Actor | None |
| Description | Allow user to login and register. |
| Pre-condition | The user has registered account. |
| Post-condition | The user is logged in. |
| Fail Scenario | Invalid username or password. |

Table 3.3 Use Case Description for browse Water Quality Prediction

|  |  |
| --- | --- |
| Case Name | Browse Water Quality Information |
| Primary Actor | Authenticated User |
| Secondary Actor | None |
| Description | Allow users to easily browse water quality information on the internet |
| Pre-condition | The user is logged in |
| Post-condition | The user can access and explore water quality information |
| Fail Scenario | None |

Table 3.4 Use Case Description for Inserting Parameters

|  |  |
| --- | --- |
| Case Name | Insert Parameters |
| Primary Actor | Authenticated User |
| Secondary Actor | None |
| Description | Enable users to input specific parameters for water quality prediction |
| Pre-condition | The user is logged in |
| Post-condition | The user has submitted parameters for prediction |
| Fail Scenario | Invalid parameters - System notifies the user of issues and prompts for correction |

Table 3.5 Use Case Description for Parameter Validation

|  |  |
| --- | --- |
| Case Name | parameter validation |
| Primary Actor | Authenticated User, Admin |
| Secondary Actor | None |
| Description | Validate user-inputted parameters to ensure accuracy and prevent errors |
| Pre-condition | he user has submitted parameters for prediction |
| Post-condition | Parameters are validated, and the user receives feedback |
| Fail Scenario | Invalid parameters |

Table 3.6 Use Case Description for Viewing Prediction

|  |  |
| --- | --- |
| Case Name | View Predictions |
| Primary Actor | Authenticated User, Admin |
| Secondary Actor | None |
| Description | Allow users to view water quality predictions based on inserted parameters |
| Pre-condition | The user has successfully submitted parameters |
| Post-condition | The user can analyze and interpret presented predictions |
| Fail Scenario | None |

**Non-Functional Requirements**

The non-functional requirements describe how the system works or that specifies criteria that can be used to judge the operation of a system. It also describes system attributes security, performance, maintainability, scalability, and usability.

The non-functional requirements of our system are:

* ﻿﻿The system is available 24\*7.
* ﻿﻿System uses hashing for the password in login. No one can know the password as the password is encrypted.
* The system complies with relevant regulatory standards and guidelines governing water quality assessment and prediction, ensuring adherence to best practices and legal requirements.

### **3.1.2 Feasibility study**

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. The feasibility analysis is divided into four parts as described below:

**Technical Feasibility:**

Technical analysis is concerned with determining how possible a system is from a technical perspective. The project is developed for general use. To access this website, the user needs an internet connection. The main requirement of the system from a developer's view is a web server capable of handling the content, internet connection, and manpower to handle the website.

**Operational Feasibility:**

It is concerned with the operating capabilities of the system. Since it is a web-based application, it is quite easy to handle the system with a normal web surfing skill. For the efficient operation, only a general-purpose computer is required. And the user interface is friendly. Hence, the system is feasible operationally.

**Schedule Feasibility:**

The term members were responsible for different aspects of the system development. The project was intended to be completed within 50-60 days; it was feasible with respect to time also. The schedule of the product is presented in the Gantt Charts below which describes the time specified for different tasks performed during system development.

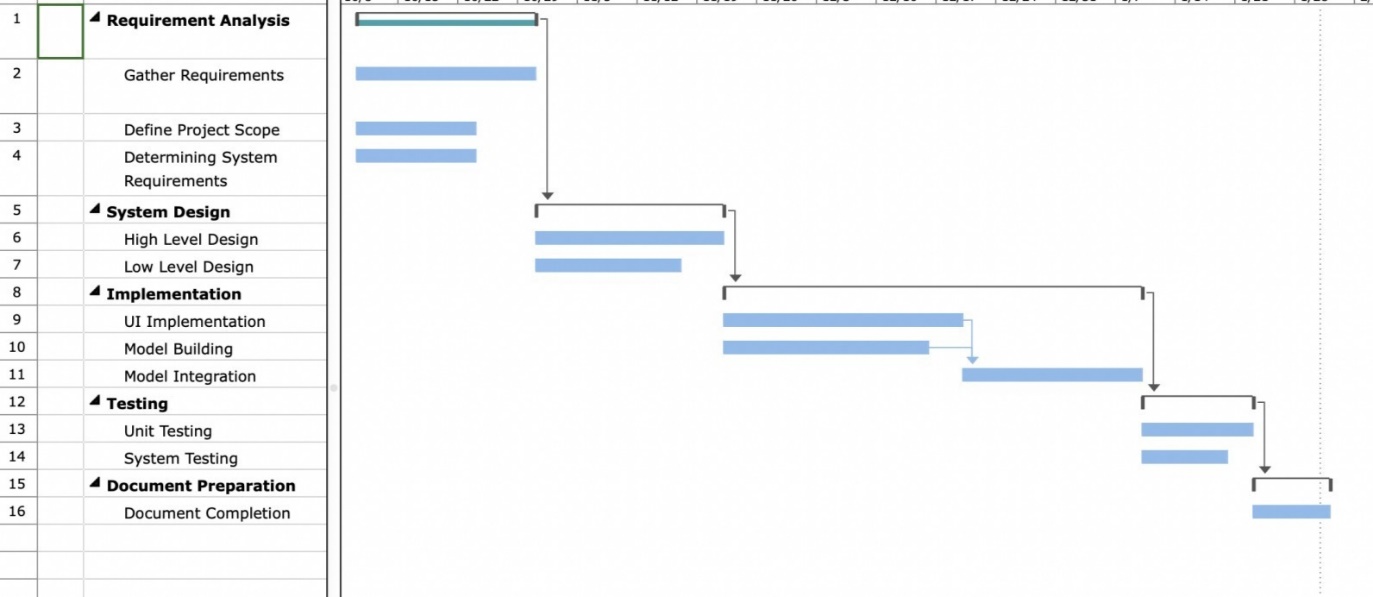


Figure 3.2 Gantt Chart

### **Process Analysis**

**Entity Relationship Diagram**

The Entity-Relationship model (ER model) for the water quality prediction system is illustrated through an Entity Relationship Diagram (ER Diagram), serving as a pivotal design framework for the database. The major entities in this ER diagram encompass the User, Prediction Parameters, Water Quality Data, Prediction Results, and Session Log. The User entity includes attributes like User ID, Username, and Password, representing registered users. Prediction Parameters capture user-inputted parameters for predictions. Prediction Results encompass the outcomes of predictions. Session Log tracks user activities within the system. The ER diagram visually portrays the system's database structure, acting as a foundational guide for implementing the water quality prediction system.

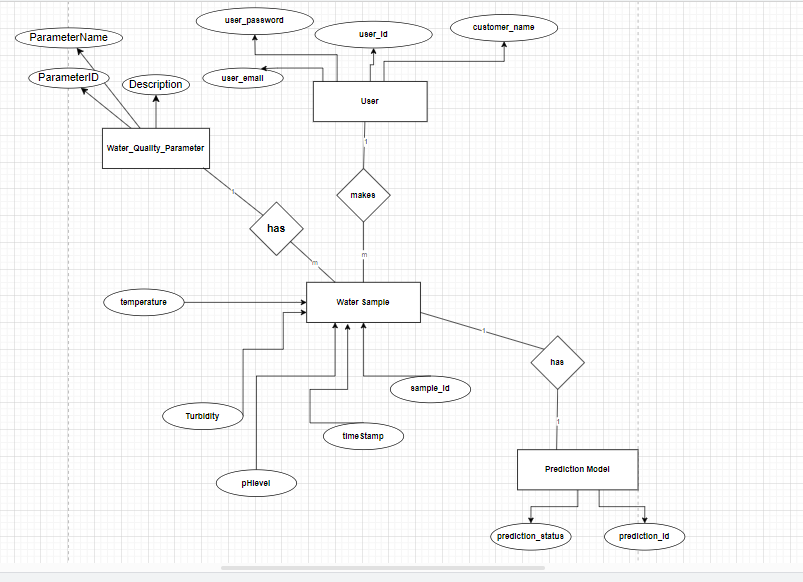


Figure 3.3 ER- Diagram

# CHAPTER 4 - SYSTEM DESIGN

## **4.1 Design**

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of product development.

### **4.1.1 Class Diagram**

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system. The class diagram is also considered as the foundation for component and deployment diagrams.

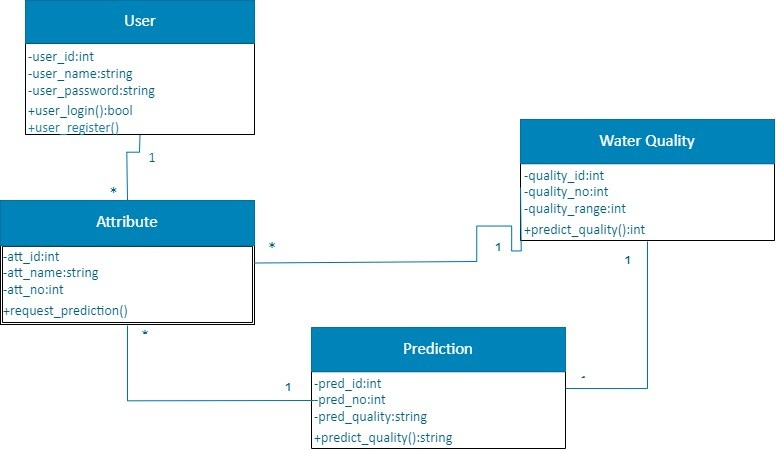


Figure 4.1 Class Diagram

### **4.1.2 Activity Diagram**

We use Activity Diagrams to illustrate the flow of control of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we depict workflows visually using an activity diagram. An activity diagram focuses on the condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

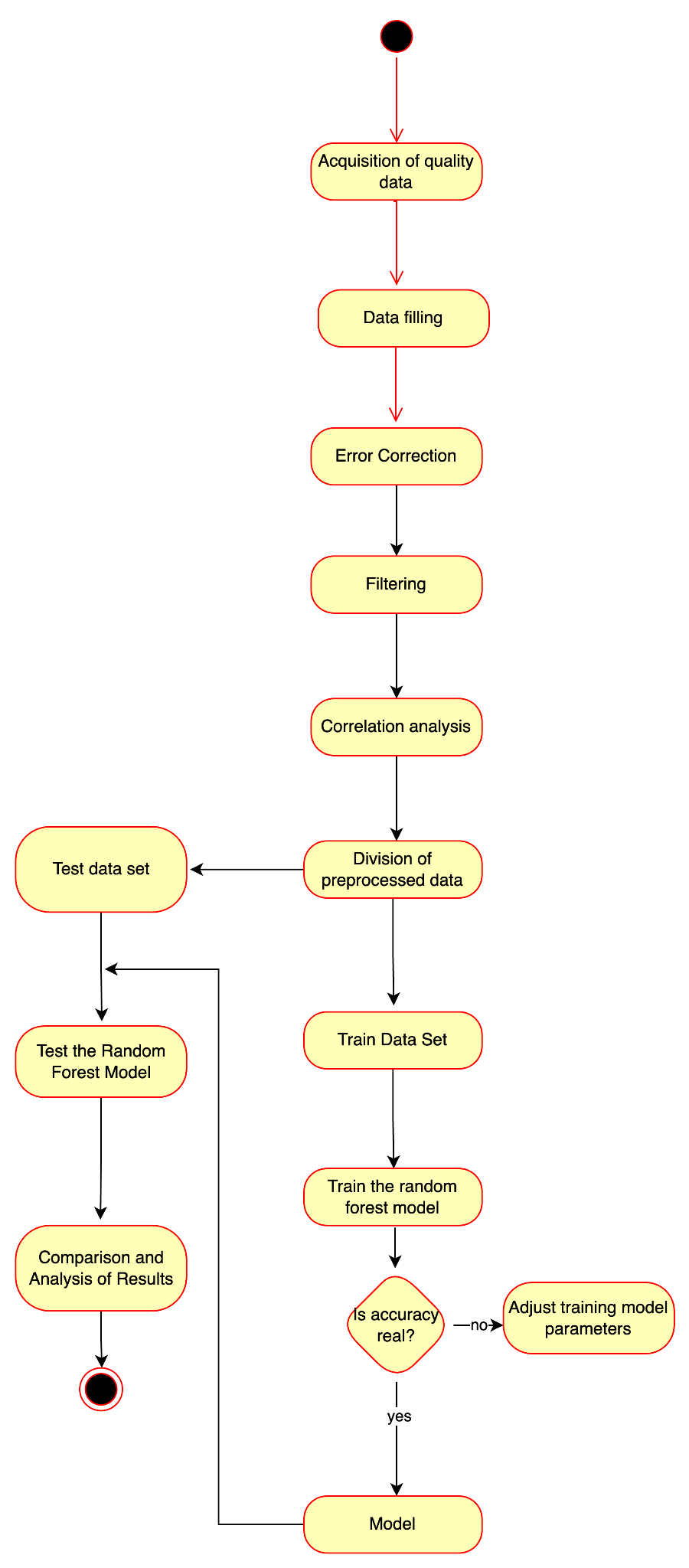


Figure 4.2 Activity Diagram

### **4.1.3 Sequence Diagram**

A sequence diagram is a type of interaction diagram because it describes how- and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. The sequence diagrams of this system are drawn as shown below:

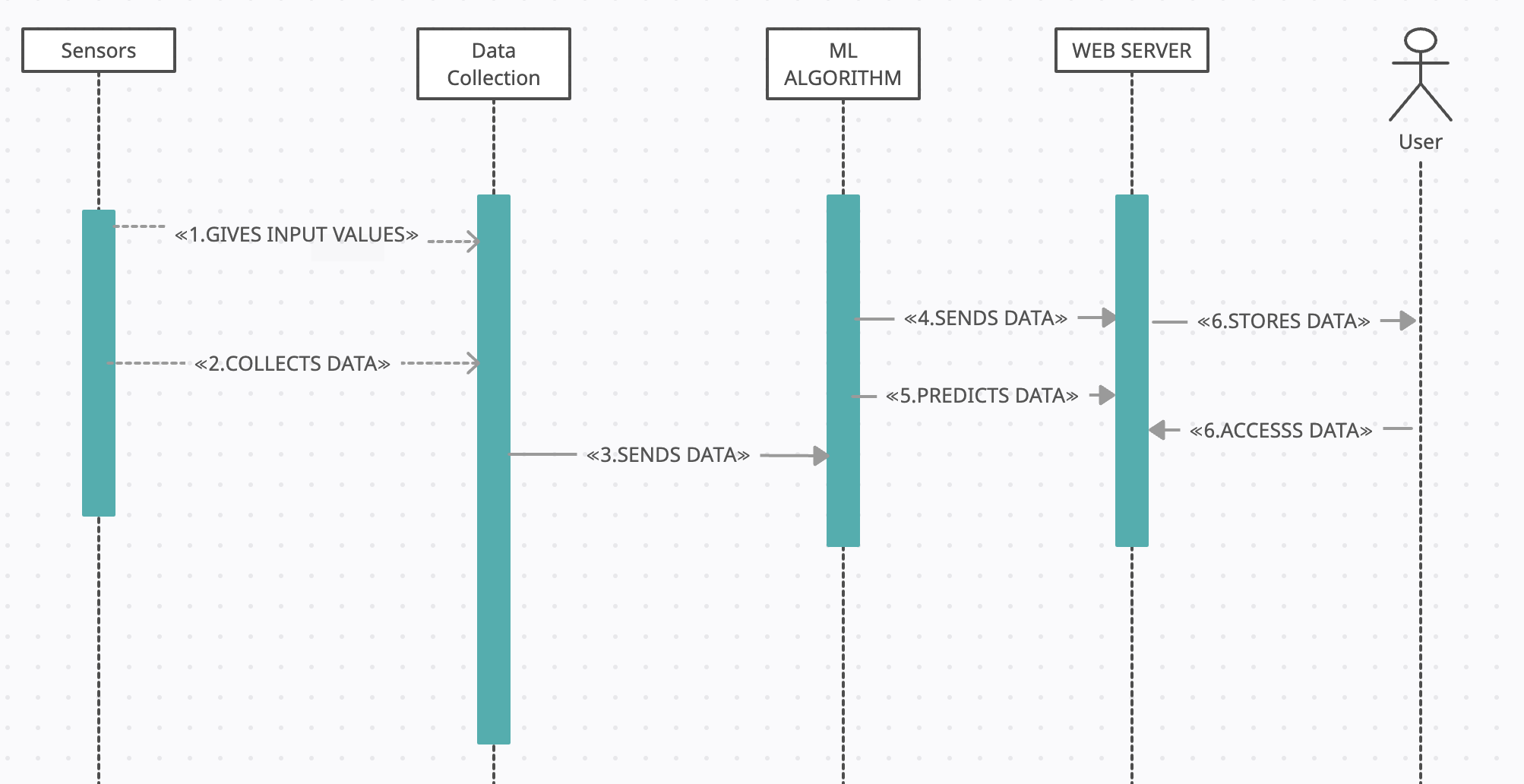


Figure 4.3 Sequence Diagram

## **4.2 Algorithm Details**

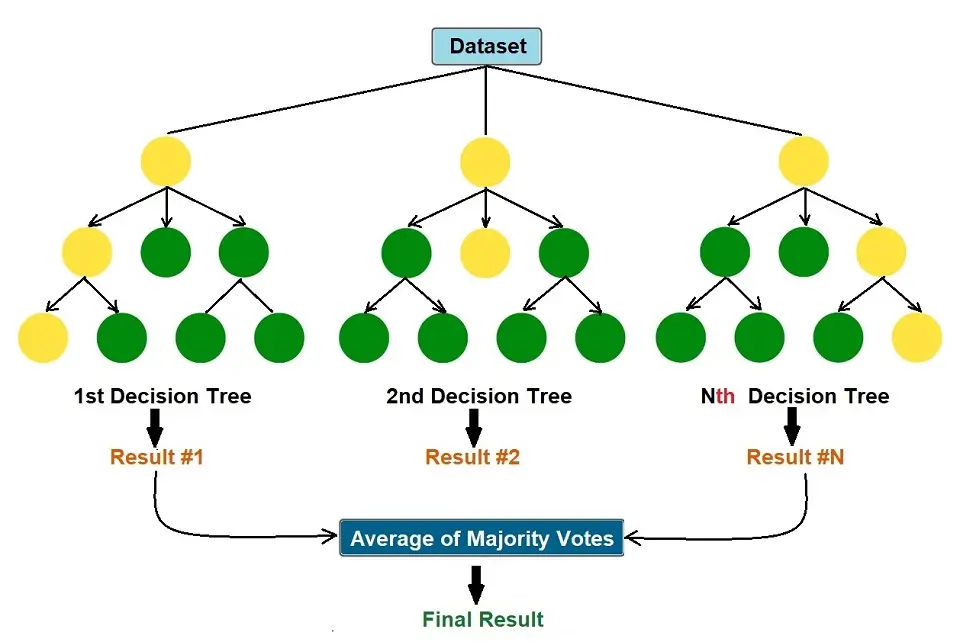
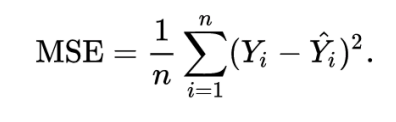
The random forest algorithm is a widely used supervised algorithm consisting of many decisions trees. The algorithm establishes the outcome based on the predictions of the decision trees. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, it predicts the final output. A simple representation of the model architecture is shown below:

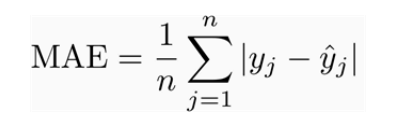
Figure 4.4 Random Forest Algorithm

As the models used for this research are regression models, the evaluation metrics selected for this research are Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, and R-squared. These evaluation metrics were chosen because of their suitability for regression algorithms and wide use within the wider research area and the literature reviewed in Section 2 of this research paper. Each evaluation method is discussed in more detail below:

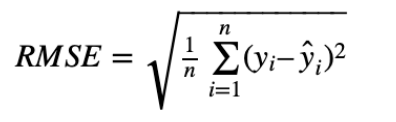
Mean Squared Error (MSE) - The MSE calculates how close a regression line is to the data points. It is calculated by subtracting the predicted value from the observed value and squaring that difference. A small MSE is preferred as it shows that the error in the model is small (Gupta, 2022).



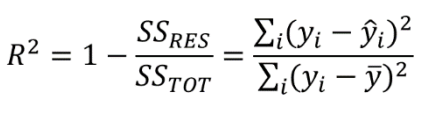
Mean Absolute Error (MAE) is calculated by looking at the mean over the absolute differences between observed and predicted values. Both directions are treated the same which means that outliers do not play a role (MAE, 2022).



Root Mean Squared Error (RMSE) shows how far the prediction values are from the true values using the Euclidean distance. As a rule of thumb, the lower the RMSE the better the model fits the data (Grace-Martin, 2022)



R-squared (R2) shows how well the model fits the data. If the R2 is close to 1, it shows that the model has predicted well. If the model is close to 0, the larger the distance between the actual and predicted values (R2, 2022).



# CHAPTER 5- IMPLEMENTATION AND TESTING

## **5.1 Implementation**

### **5.1.1 Analysis and Design Tools**

In this stage, physical system specification is converted into a working and reliable solution. This is where the system is developed. On receiving the system design documents, the work is divided into modules/units and actual coding is started. It is followed by testing. Several tools and algorithms are used in this phase of software development.

The design tools for creating different figures such as Entity Relationship diagrams, use-case, flow-chart, sequence diagrams, and other required diagrams can be designed through draw.io, MS Visio, and MS Project. The produced facts and figures are analyzed to release the information that is required to maintain the transaction in the application. This tends to process the working mechanism and can also be used for future work.

### **5.1.2 Implementation Tools (Frontend and Backend)**

**5.1.2.1 HTML, CSS, JavaScript for Frontend**

In this application HTML, CSS, and JavaScript are used for creating frontend. HTML provides a basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript. CSS is used to control presentation, formatting, and layout whereas JavaScript is used to control the behaviour of different elements.

**5.1.2.2 Flask Framework for Backend**

In this application Flask Framework is used for the backend. Flask is a high-level Python web framework that enables the rapid development of secure and maintainable websites. Built by experienced developers. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

**5.1.2.3 SQLite Database for Database Management**

SQLite is an in-process library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of SQL query language. It’s also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

### **5.1.3 Description of the Algorithm**

**5.1.3.1 Random Forest Algorithm**

A Random Forest is an ensemble learning algorithm that belongs to the family of decision tree algorithms. It is used for both classification and regression tasks. The main idea behind Random Forest is to build multiple decision trees during training and combine their outputs to improve overall performance and reduce overfitting.

**Steps of Random Forest Algorithm**

Data Collection: Collect a dataset with features and corresponding target values. Ensure the dataset is representative of the problem you are trying to solve.

Data Pre-processing: Handle missing data: Impute missing values or remove incomplete records. Encode categorical variables if necessary. Normalize or scale numerical features to bring them to a similar scale.

Data Splitting: Split the dataset into training and testing sets. The training set is used to train the model, and the testing set is used to evaluate its performance.

Random Forest Model Creation: Choose the number of trees to include in the forest.

For each tree: Perform bootstrapped sampling (random sampling with replacement) from the training data. Randomly select a subset of features at each split point. The number of features can be controlled by the max features parameter. Build a decision tree based on the bootstrapped sample and the selected features. Train multiple trees to form the forest.

Model Training: Train the Random Forest model on the training dataset using the ensemble of decision trees.

Model Evaluation: Evaluate the performance of the Random Forest model on the testing set using appropriate metrics (e.g., accuracy, precision, recall, F1 score, or Mean Squared Error for regression). Utilize techniques like cross-validation to ensure robust evaluation.

Feature Importance Analysis: Analyze feature importance scores provided by the Random Forest model. This helps identify which features contribute the most to the model's predictions.

Hyper parameter Tuning: Fine-tune hyper parameters, such as the number of trees (n estimators), maximum tree depth (max depth), minimum samples per leaf (min samples leaf), etc. This can be done using techniques like grid search or randomized search.

Optimization and Fine-Tuning: Iterate on the model, adjusting hyper parameters or features as needed based on the performance evaluation.

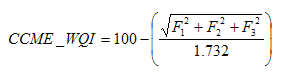
Prediction: Once satisfied with the model performance, use the trained Random Forest to make predictions on new or unseen data.

Model Interpretability: Random Forests offer some level of interpretability through feature importance analysis. Interpret the results to gain insights into the factors influencing predictions.

Monitoring and Maintenance: Continuously monitor the model's performance over time. Periodically retrain the model using new data to ensure it stays relevant.

**WQI calculation**

**CCME Water Quality Index formula**

****

Calculation of the index is based on three terms: **scope** (F1) – number of parameters that are not compliant with the water quality guidelines, **frequency** (F2) – number of times that the guidelines are not respected and **amplitude** (F3) – the difference between non-compliant measurements and the corresponding guidelines.

The division of these terms by 1.732 is based on the fact that each of the three factors contributing to the index can reach 100. The maximal length is, therefore, expressed as

Formula 2

Division by 1.732 reduces the maximal length to 100. The index produces a value from 0 to 100. The higher the number, the better the water quality.

**Explanation of each term of the index**

First of all, **the term F1** (scope) expresses the percentage of parameters for which at least one measurement did not comply with the corresponding guideline during the period under study:

Formula 3

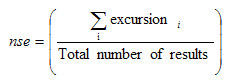
**The term F2** (frequency) represents the percentage of analytical results that do not comply with the guidelines.

Formula 4

**Finally, the term F3** (amplitude) represents the *difference* between the non-compliant analytical results and the guidelines to which they refer. The term F3 is an asymptotic function, representing the normalized sum of excursions (NSE) guidelines within the range of values from 0 to 100.

Formula 5

To calculate the overall degree of non-compliance, we add the excursions of non-compliant analytical results and divide the sum by the total number of analytical results. This variable is called the normalized sum of excursions (NSE).



There are three possible ways of determining the excursion:

* If the finding must not exceed the guideline:  
  Formula 7
* If the finding must not be lower than the guideline:  
  Formula 8
* If the guideline is zero (equal to zero):  
  Formula 9

The appendix contains a specific example of explaining the index calculation.

**Index value categorization**

Once the index has been calculated, we obtain a value of 0 to 100. The higher the index value, the better the water quality. The index is then placed in one of the following water quality categories:

**Excellent:** (CCME WQI value from 95.0 to 100.0) Water quality is intact. Conditions are very close to natural or desired levels. These index values can only be obtained if all measurements comply with the guidelines almost all the time.

**Good:** (CCME WQI value from 80.0 to 94.9) Water quality is intact and only one minor threat or deterioration is observed; conditions rarely differ from the natural or desirable levels.

**Fair:** (CCME WQI value from 65.0 to 79.9) Water quality is usually intact, but occasionally endangered or deteriorated; conditions sometimes deviate from the natural or desirable levels.

**Marginal:** (CCME WQI value of 45.0 to 64.9) Water quality is frequently endangered or deteriorated; conditions often deviate from the natural or desirable levels.

**Poor:** (CCME WQI value from 0.0 to 44.9) Water quality is almost always endangered or deteriorated; conditions usually deviate from natural or desirable levels.

**Implementation Details of Module**

**Description:**

This module implements the Decision Tree and Random Forest algorithms for water quality classification. The Decision Tree algorithm is used to create a tree-like model of decisions based on features of water quality parameters data, while the Random Forest algorithm combines multiple decision trees to improve prediction accuracy. The module includes functions to train the models, make predictions, and evaluate their performance.

Table 5.1 Model Accuracy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | R-Squared | MAE | MSE | Execution Time |
| **Random Forest** | **0.48** | **0.18** | **0.58** | **8.44 seconds** |
| **Decision Tree** | **0.42** | **0.21** | **0.65** | **0.74 seconds** |

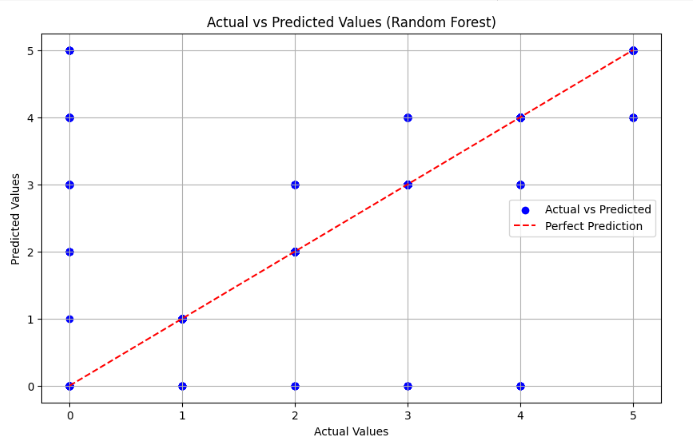
****

Figure 5.1 Result using Random Forest

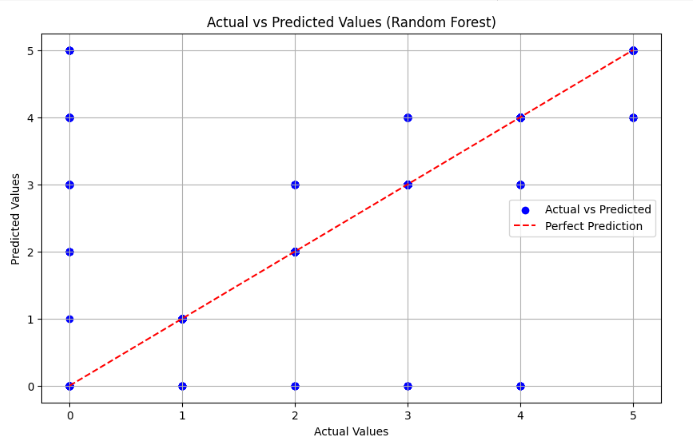
****

Figure 5.2 Result Using Decision Tree

## **5.2 Testing**

### **5.2.1 Test Plan**

**Introduction**

A test plan for water quality prediction involves outlining the steps and procedures to assess the performance, accuracy, and reliability of the predictive model.

**In Scope**

Testing includes:

1. Unit Testing

2. Integration Testing

3. System Testing

**Unit Testing**

Unit testing is a software testing method by which individual units of source code, sets of one or more computer modules together with associated control data, usage procedures, and operating procedures are tested to determine whether they are fit for use. In this system, the individual units are individually tested by providing one or few inputs and expected for the single output.

**Integration Testing**

Integration testing for water quality prediction involves ensuring that different components of the prediction system work together seamlessly and produce accurate and reliable predictions.

### **5.2.2 Unit Test Cases**

Table 5.2 Test Case for Register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Description** | **Prerequisite** | **Steps** | **Input** | **Expected Result** | **Actual Result** |
| 1 | Verify if a user can register with valid data. | The user should be on the register page. | 1. Enter First Name  2. Enter Last Name  3. Enter Email Address  4. Enter Username  5. Enter Password  6. Enter Confirm Password | 1.FirstName= Test  2.LastName=  Ing  3. Email Address= **sneha@gmail.com**  4.Username=  Test user  5. Password=  Sneha789  6. Confirm Password =  test@123 | The user should be logged in and directed to the chat page. | The user is logged in and directed to the chat page. |
| 2 | Verify if a user can register with blank inputs. |  |  | Leave either one, more, or all the fields empty and submit. | The System should display an error message saying a particular field cannot be blank. | The system displayed an error message saying a particular field was left blank. |

Table 5.3 Test Case for Login Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Description** | **Pre-requisite** | **Steps** | **Input** | **Expected Result** | **Actual Result** |
| 1 | Verify if a user can log in with the correct email and password. | The user should be on the Login Page. | 1. Input email  2.Input Password  3. Click Login | Email = **sneha@gmail.com**  Password = Sneha789 | The user should be logged in. | The user logged into the home page. |
| 2 | Verify if a User can log in with an incorrect email and password. |  |  | Email = **sneha@gmail.com**  Password = Sneha@789 | The system should display an error saying “Incorrect username or password”. | The system displayed an error message. |
| 3 | Verify if the user can log in with an empty email and password. |  |  | Either of the two fields is blank or all fields are blank. | The user shouldn’t log in. | The user couldn’t log in. |

Table 5.4 Test Case for User Prediction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N** | **Description** | **Prerequisite** | **Steps** | **Input** | **Expected Result** | **Actual Result** |
| 1 | Good Quality | Water quality model trained | Gather input parameters. | - pH Level: 7.2  - Dissolved Oxygen: 8.5 mg/L  - Temperature: 22°C | Good | Good |
| 2 | Fair Quality | Water quality model trained | Gather input parameters. | - pH Level: 6.5  - Dissolved Oxygen: 5.2 mg/L  - Temperature: 18°C | Fair | Fair |
| 3 | Poor Quality | Water quality model trained | Gather input parameters. | - pH Level: 8.0  - Dissolved Oxygen: 4.0 mg/L  - Temperature: 25°C | Poor | Poor |
| 4 | Excellent Quality | Water quality model trained | Gather input parameters. | - pH Level: 7.8  - Dissolved Oxygen: 9.0 mg/L  - Temperature: 20°C | Excellent | Excellent |

**5.2.3 System Testing**

System testing for water quality prediction involves evaluating the entire prediction system as a whole to ensure that it meets the specified requirements and functions correctly in its intended environment.

Table 5.5 System Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Case | Test Inputs | Expected result | Status |
| 1. | The user enters the login page | Email: Sneha@gmail.com  Password: Sneha789 | Display homepage with user logged in | Homepage with user logged in status is displayed. |
| 2. | User clicks the prediction button | Homepage: Surfing (Scrolling, view try prediction) | Display information about water test with try prediction button | prediction page is displayed |
| 3. | The user enters the parameters | WQI Fecal:91  WQI pH:95  WQITemperature:81  WQI Phosphorus:84  WQI Oxygen:85  WQI Total Sediment:78  WQI Nitrogen:99  WQI Turbidity:69 | Display result “Bad”.  “You water needs further purification and testing”. | Water quality result is displayed. |
|  |  |  |  |  |

# CHAPTER 6- CONCLUSION AND ENHANCEMENTS

## **6.1 Limitations of Project**

* User input affects the accuracy.
* Complexity of Environmental Systems.

## **6.2 Conclusions**

The water quality prediction model that can be used in the field of water quality monitoring. Using machine learning and a Random forest classifier for collecting, storing, and analyzing water samples is a much more effective and efficient method for water quality evaluation than regular laboratory tests. This motivated us to create a machine learning model based on the Random Forest algorithm to evaluate the quality of river water based on 8-distinctive features: pH, hardness, presences of solids, presence of chloramines, presence of sulfates, conductivity, organic carbon, trihalomethanes, turbidity, and finally portability.

A complete website with a user-interactive interface. A study on water quality prediction using the Random Forest algorithm would typically involve summarizing key findings, assessing the performance of the model, and providing insights into the potential applications and limitations of the approach. The quality of water that we use for drinking or cooking has a direct effect on our health, which is why having perfectly safe water is not only a right for humans but also extremely critical.

The project has been developed on Flask Framework with HTML, CSS, and JavaScript as frontend tools. It is a very important site and may have a good future coming.

## **6.3 Future Enhancements**

A water quality prediction using the Random Forest algorithm could focus on addressing existing limitations and leveraging advancements in technology and data science.

Here are some potential avenues for improvement

* Integration with Real-time Monitoring Systems.
* Automated Feature Engineering.
* Education and Outreach.

# REFERENCES

[1] S. M. Alomani, Najd Ibrahim Alhawiti, A’aeshah Alhakamy, “Prediction of Quality of Water According to a Random Forest Classifier.” International Journal of Advanced Computer Science and Applications (IJACSA), 2022.

[2] C. Gakii, Jepkoech Jeniffer, “A CLASSIFICATION MODEL FOR WATER QUALITY ANALYSIS USING DECISION TREE.” European Centre for Research Training and Development UK (www.eajournals.org), Jun. 2019.

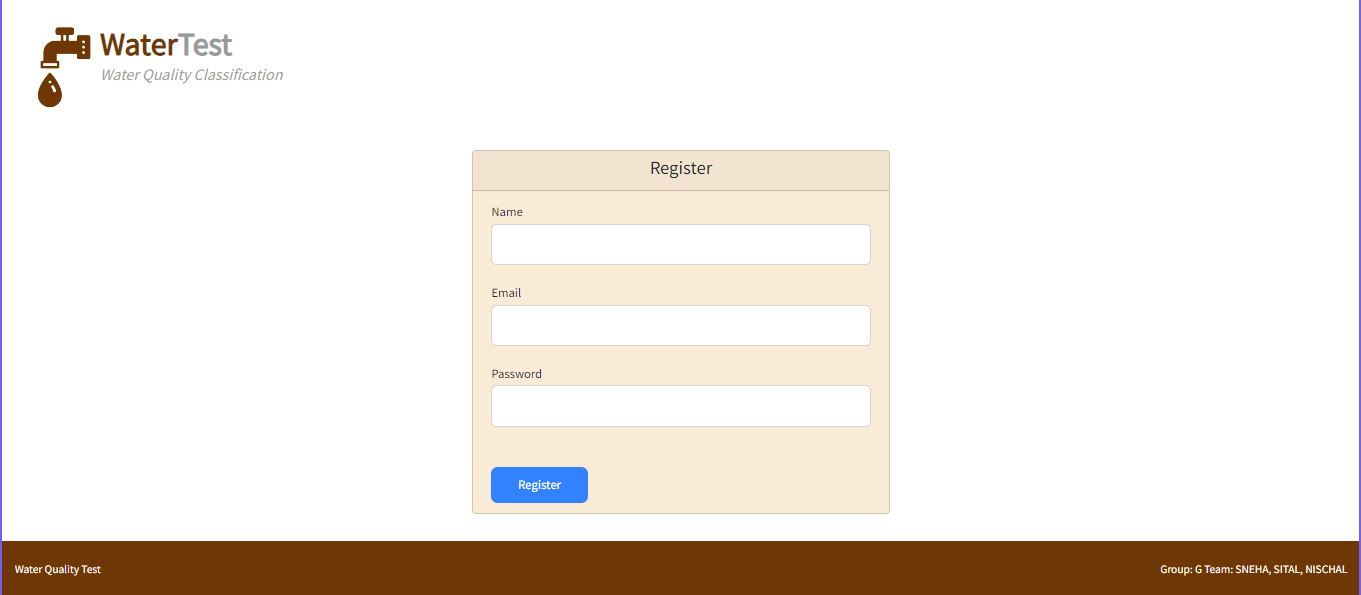
[3] A. Danades, Devie Pratama, Dian Anggraini, Diny Anggriani, “Comparison of accuracy level K-Nearest Neighbor algorithm and support vector machine algorithm in classification water quality status.” School of Electrical Engineering and Informatics, Institute Teknologi Bandung, Oct. 03, 2016.

[4] Canadian Council of Ministers of the Environment, 2001, "Canadian Water Quality Guidelines for the Protection of Aquatic Life: CCME Water Quality Index 1.0, user's manual", in *Canadian Environmental Quality Guidelines, 1999,* Winnipeg.

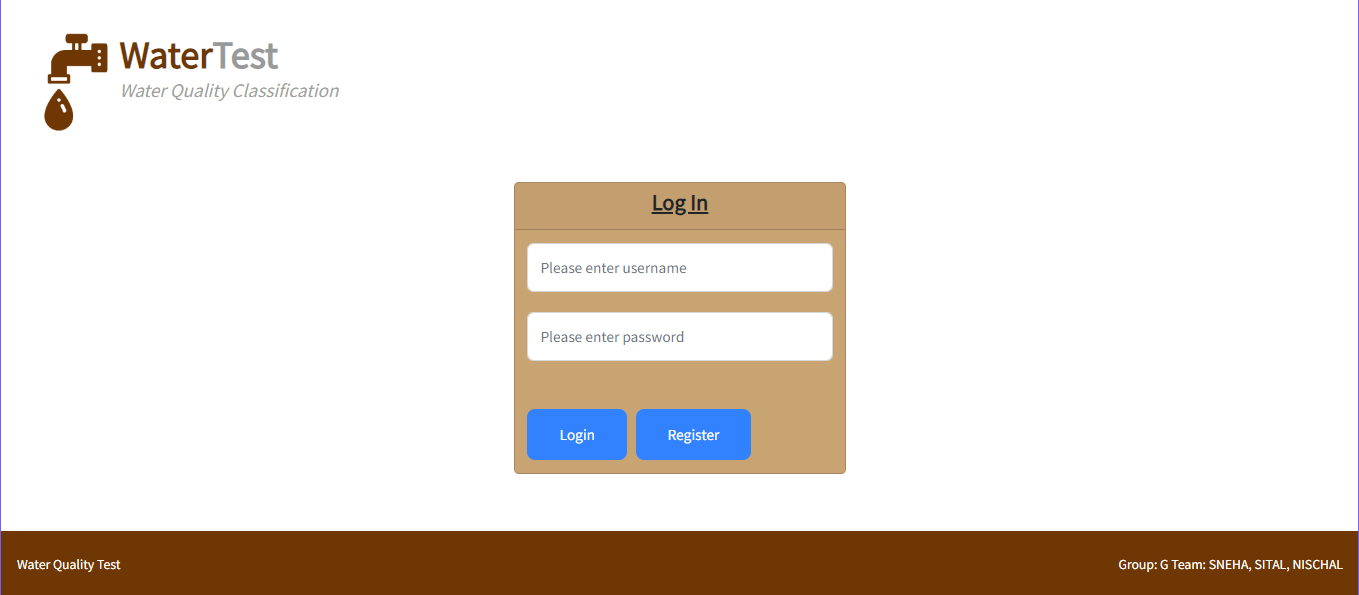
[5] H. Mohammed, I. A. Hameed, R. Seidu “Random forest tree for predicting fecal indicator organism:drinking water supply.” BESC, 2017.

# APPENDICES

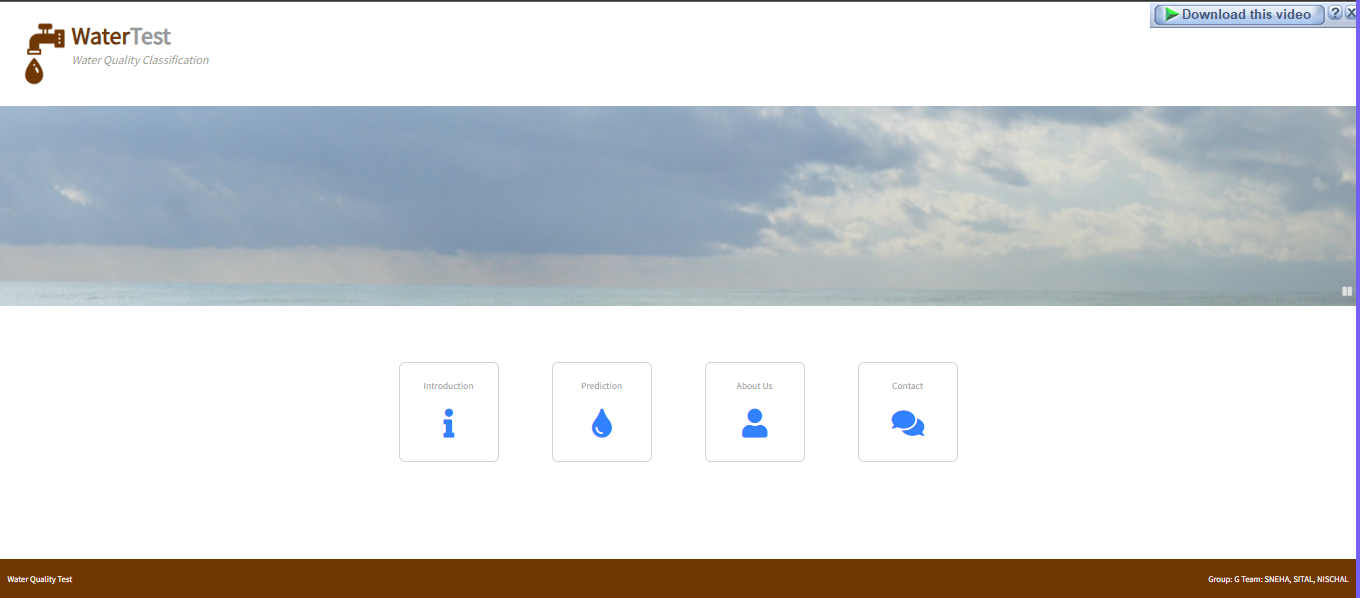
**Screenshots**

****

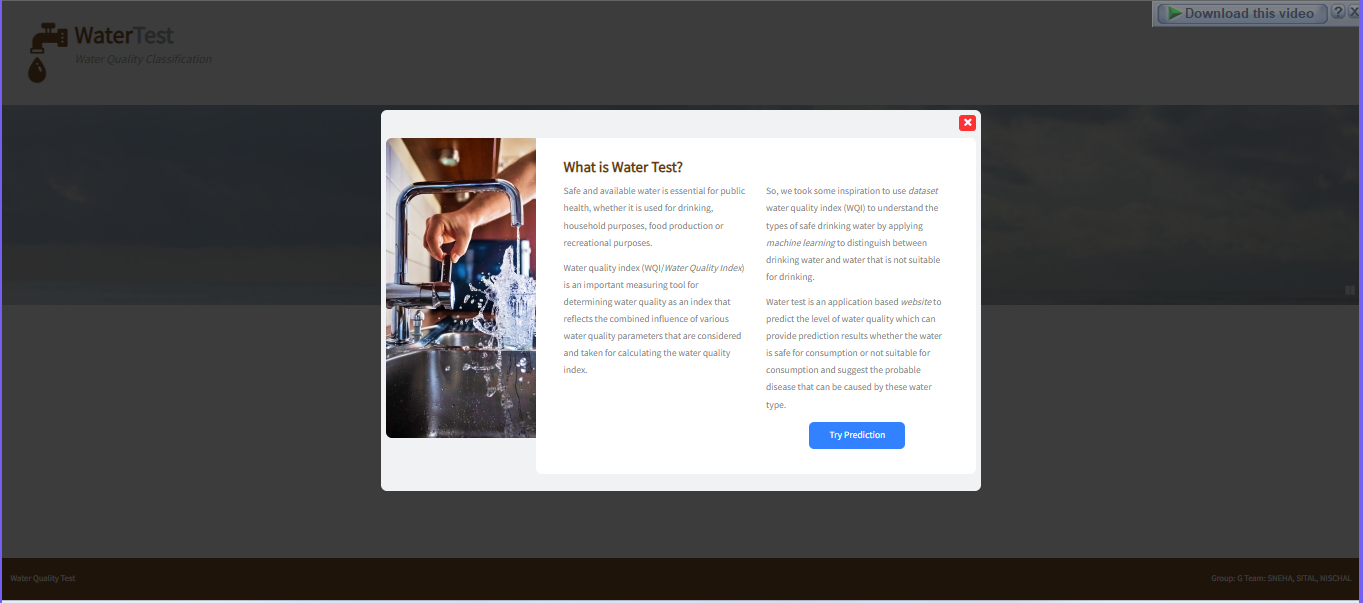
**Sign Up**

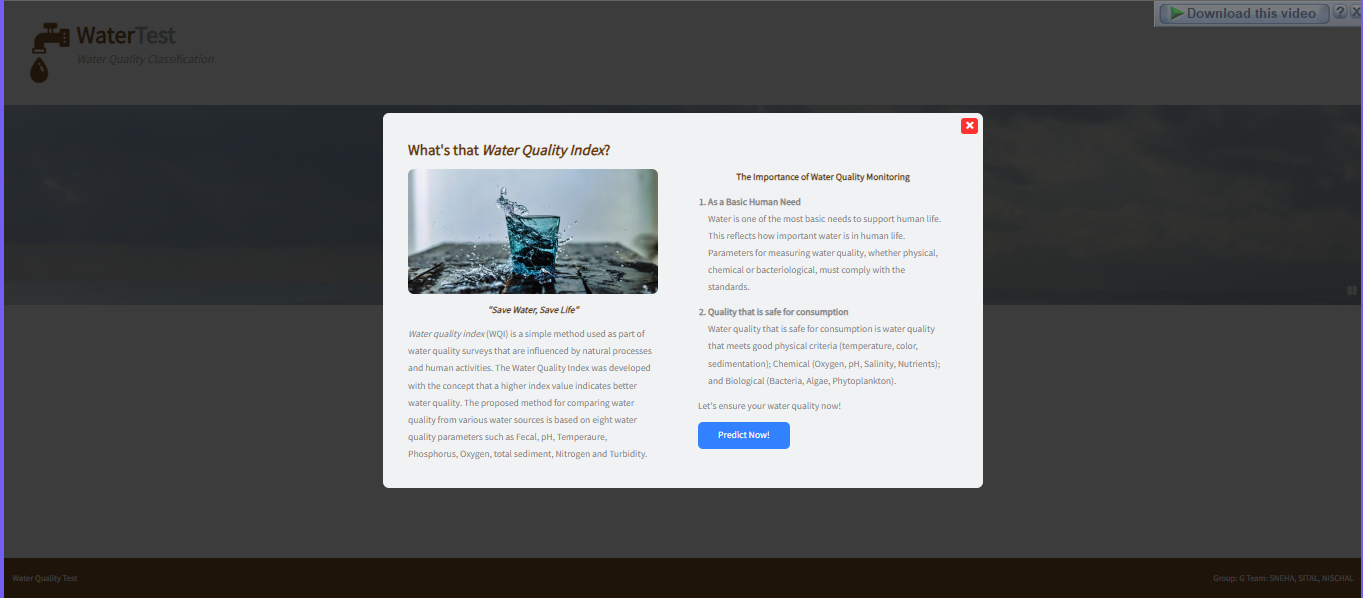
****

**Login Page**

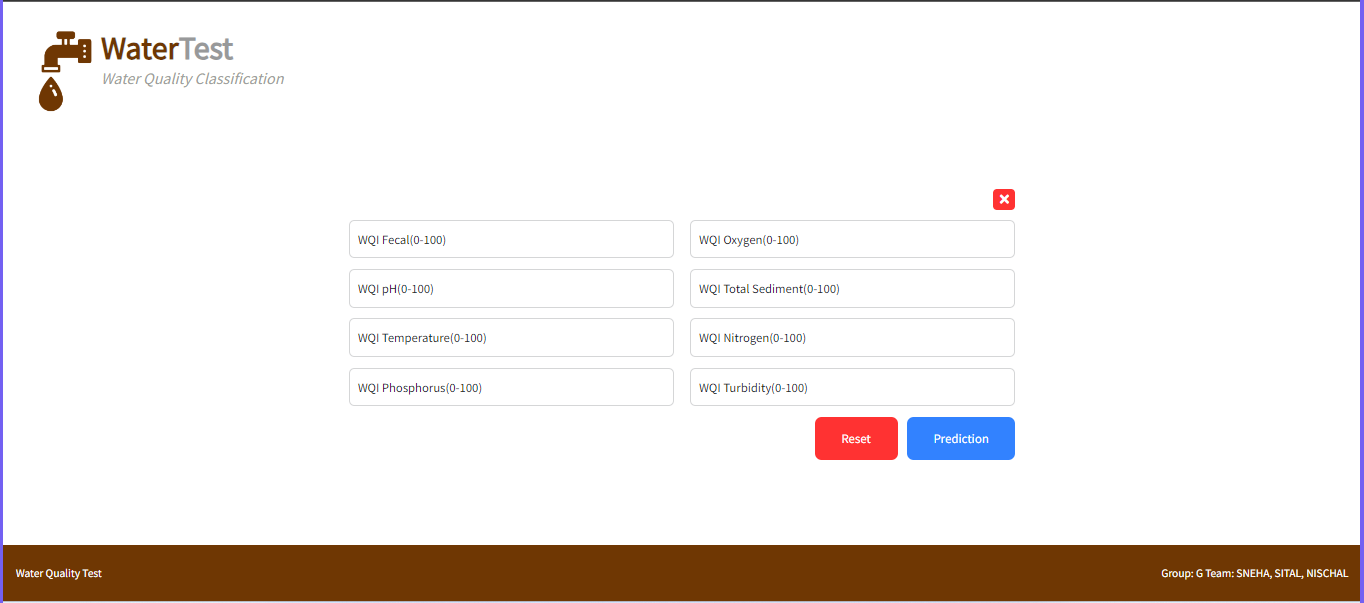
****

**Home Page**

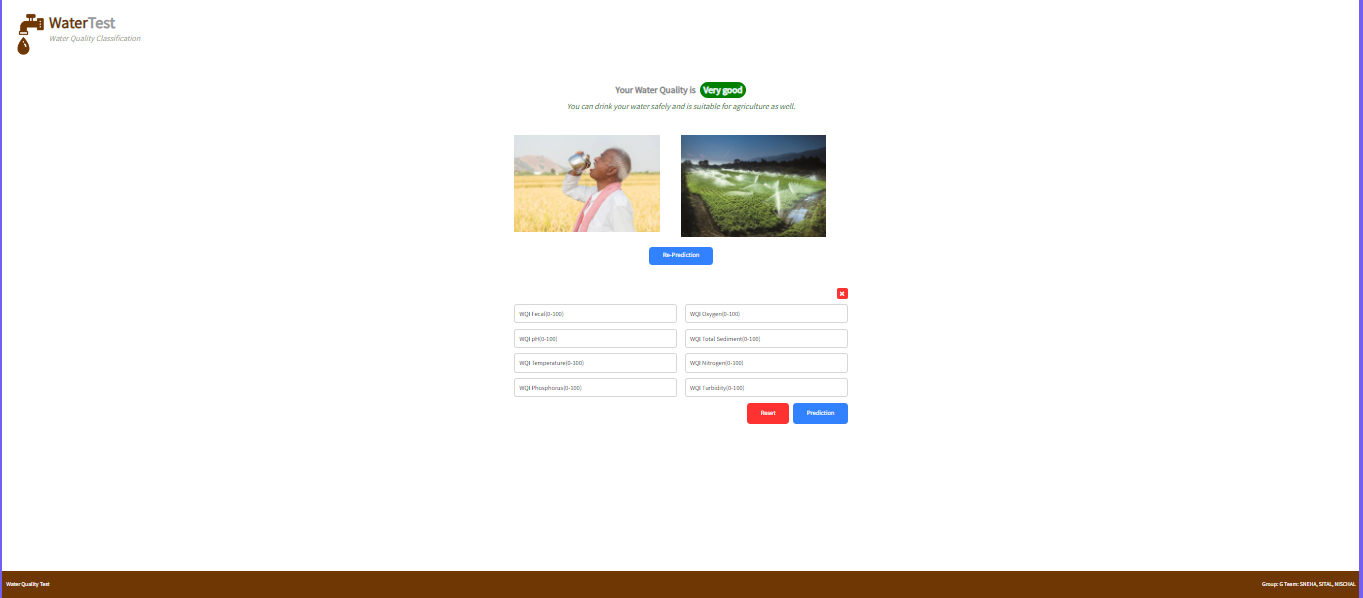
****

****

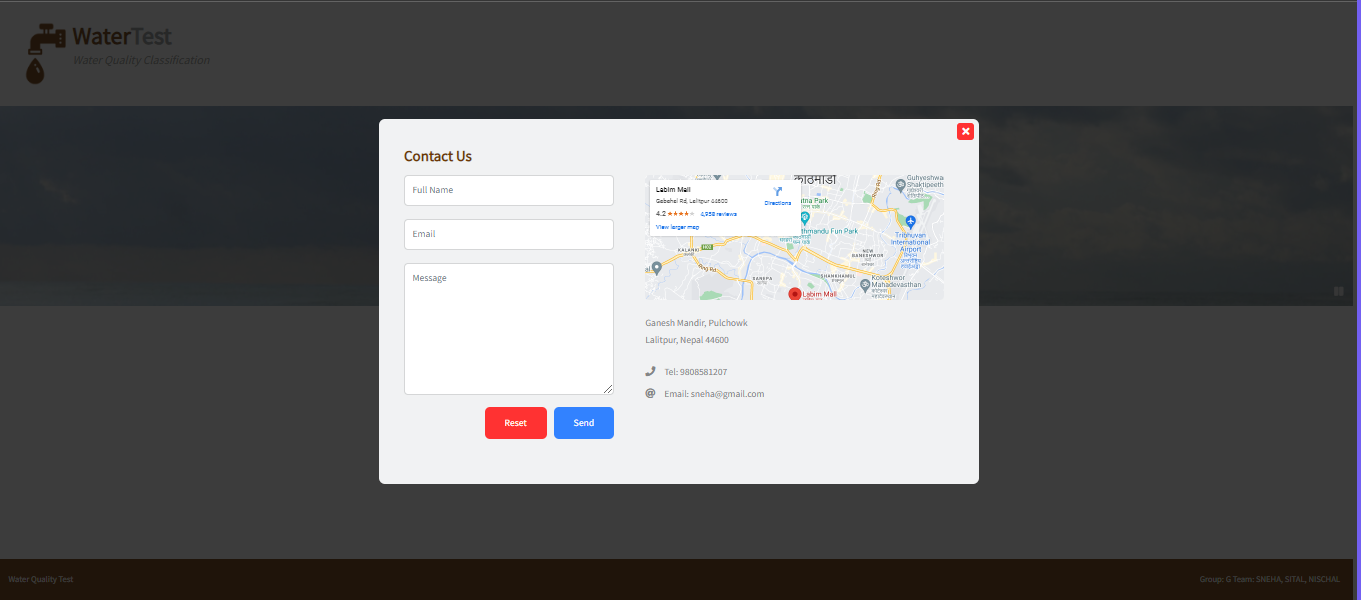
**Description**

****

**Prediction Page**

****

**Result Page**

**Contact Us:**