#### A

Mini Project On

## WATER QUALITY MONITORING AND FORECASTING SYSTEM

#### (Submitted in partial fulfillment of the requirements for the award of Degree) BACHELOR OF TECHNOLOGY

in

#### COMPUTER SCIENCE AND ENGINEERING

By

#### P. Indhu (217R1A0544)

K. Mahipal (217R1A0528)

#### E. Mahesh Kumar (217R1A0520)

Under the Guidance of

**Ms. TABEEN FATIMA**

(Assistant Professor)



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2021-25**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

This is to certify that the project entitled “**WATER QUALITY MONITORING AND FORECASTING SYSTEM**” being submitted by **P. Indhu (217R1A0544), K. Mahipal (217R1A0528) & E. Mahesh Kumar (217R1A0520)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Ms. Tabeen Fatima Dr. A. Raji Reddy**

**Assistant Professor DIRECTOR**

**INTERNAL GUIDE**

**Dr. Nuthanakanti Bhaskar EXTERNAL EXAMINER HOD**

**Submitted for viva voice Examination held on**

**ACKNOWLEGDEMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express our profound gratitude and deep regard to our guide **Ms. Tabeen Fatima,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. K. Maheswari, Dr. J. Narasimha Rao, Ms. K. Shilpa, Mr. K. Ranjith Reddy** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. Nuthanakanti Bhaskar,** Head, Department of Computer Science and Engineering for providing encouragement and support for complete this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**P. INDHU (217R1A0544)**

**K. MAHIPAL (217R1A0528)**

**E. MAHESH KUMAR(217R1A0520)**

### ABSTRACT

Water pollution refers to the release of pollutants into the water that is detrimental to human health and the planet as whole. The aim is to investigate machine learning-based techniques for water quality forecasting by predicting results with the best accuracy. The analysis of the data set by supervised machine learning technique(SMLT) to capture information like variable identification, uni-variate analysis, bi-variate and multivariate analysis, missing value treatments and analysis data validation, data cleaning/preparation, and data visualization will be done on the entire given data set. Our analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to performance in the prediction of water quality pollution by accuracy calculation. To propose a machine learning-based method to accurately predict the Water Quality Index value by prediction results in the form of best accuracy from comparing supervised classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms from the given transport traffic department data set with evaluation classification report, identify the confusion matrix and categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with the best accuracy with precision, Recall and F1 Score.

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | Project Architecture | 11 |
| Figure 3.2 | Use case diagram | 12 |
| Figure 3.3 | Class diagram | 13 |
| Figure 3.4 | Sequence diagram | 15 |
| Figure 3.5 | Activity diagram | 16 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO**. |
| Screenshot 5.1 | Copy content from DB.txt file and paste in MYSQL | 29 |
| Screenshot 5.2 | Water Quality Monitoring and Forecasting System login page | 30 |
| Screenshot 5.3 | New User Signup Page | 30 |
| Screenshot 5.4 | Signup Process Completed | 31 |
| Screenshot 5.5 | User Login Screen | 31 |
| Screenshot 5.6 | Click on Load and Preprocess Data | 32 |
| Screenshot 5.7 | Water Quality Graph | 32 |
| Screenshot 5.8 | Data processed and loaded | 33 |
| Screenshot 5.9 | LSTM Algorithm output | 34 |
| Screenshot 5.10 | Random Forest Algorithm output | 34 |
| Screenshot 5.11 | Upload test data file | 35 |
| Screenshot 5.12 | Water Quality Forecasting Result | 36 |

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABSTRACT** | |  |  | i |
| **LIST OF FIGURES** | | |  | ii |
| **LIST OF SCREENSHOTS** | | | | iii |
| **1.** | **INTRODUCTION** | | | 1 |
|  | 1.1 | PROJECT SCOPE | | 2 |
|  | 1.2 | PROJECT PURPOSE | | 2 |
|  | 1.3 | PROJECT FEATURES | | 3 |
| **2.** | **SYSTEM ANALYSIS** | | | 4 |
|  | 2.1 | PROBLEM DEFINITION | | 5 |
|  | 2.2 | EXISTING SYSTEM | | 6 |
|  |  | 2.2.1 | LIMITATIONS OF THE EXISTING SYSTEM | 6 |
|  | 2.3 | PROPOSED SYSTEM | | 7 |
|  |  | 2.3.1 | ADVANTAGES OF PROPOSED SYSTEM | 7 |
|  | 2.4 | FEASIBILITY STUDY | | 7 |
|  |  | 2.4.1 | ECONOMIC FESIBILITY | 8 |
|  |  | 2.4.2 | TECHNICAL FEASIBILITY | 8 |
|  |  | 2.4.3 | SOCIAL FEASIBILITY | 8 |
|  | 2.5 | HARDWARE & SOFTWARE REQUIREMENTS | | 9 |
|  |  | 2.5.1 | HARDWARE REQUIREMENTS | 9 |
|  |  | 2.5.2 | SOFTWARE REQUIREMENTS | 9 |
| **3.** | **ARCHITECTURE** | | | 10 |
|  | 3.1 | PROJECT ARCHITECTURE | | 11 |
|  | 3.2 | USECASE DIAGRAM | | 12 |
|  | 3.3 | CLASS DIAGRAM | | 13 |
|  | 3.4 | SEQUENCE DIAGRAM | | 15 |
|  | 3.5 | ACTIVITY DIAGRAM | | 16 |
| **4.** | **IMPLEMENTATION** | | | 18 |
|  | 4.1  4.2  4.3  4.4 | LONG-SHORT TERM MEMORY RANDOM FOREST  DATASET  SAMPLE CODE | | 19  19  19  20 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **5**. |  | **SCREENSHOTS** | | 28 |
| **6.** |  | **TESTING** |  | 37 |
|  | 6.1 | INTRODUCTION TO TESTING | | 38 |
|  | 6.2 | TYPES OF TESTING | | 38 |
|  |  | 6.2.1 | UNIT TESTING | 38 |
|  |  | 6.2.2 | INTEGRATION TESTING | 38 |
|  |  | 6.2.3 | FUNCTIONAL TESTING | 39 |
|  | 6.3 | TEST CASES | | 40 |
| **7.** | **CONCLUSION & FUTURE SCOPE** | | | 41 |
|  | 7.1 | PROJECT CONCLUSION | | 42 |
|  | 7.2 | FUTURE SCOPE | | 42 |
| **8.** | **BIBLIOGRAPHY** | |  | 43 |
|  | 8.1 | REFERENCES | | 44 |
|  | 8.2 | WEBSITE |  | 45 |

# INTRODUCTION

## INTRODUCTION

### PROJECT SCOPE

Water Quality Monitoring and Forecasting System project encompasses a comprehensive approach to enhancing water quality management. The primary objectives are to continuously monitor key parameters such as pH, turbidity, dissolved oxygen, temperature, and contaminants while utilizing historical data to forecast future trends and potential issues. Target areas for monitoring include municipal water sources, such as rivers, lakes, and reservoirs, as well as recreational water bodies and industrial discharge points. The Water Quality Monitoring and Forecasting System project aims to leverage machine learning to enhance the monitoring and prediction of water quality parameters, ensuring safe water for consumption and environmental sustainability. This system will utilize advanced sensors to gather real-time data on key metrics like pH, turbidity, and dissolved oxygen, which will then be processed using machine learning algorithms to identify trends and predict potential water quality issues. The project will feature an intuitive web-based dashboard that visualizes data and provides alerts based on predictive analytics, allowing stakeholders to make informed decisions quickly. Key components of the scope include the deployment of sensors, development of a robust data management system, and implementation of machine learning models for forecasting, while excluding physical site construction and long-term sensor maintenance.

### PROJECT PURPOSE

Water Quality Monitoring and Forecasting System project is to safeguard public health and protect the environment by providing a comprehensive framework for monitoring and predicting water quality. By continuously tracking critical water quality parameters such as pH, turbidity, dissolved oxygen, and contaminants, the system aims to ensure that drinking water and recreational water bodies remain safe for community use. This proactive approach not only helps identify potential health risks associated with waterborne diseases but also supports environmental conservation by monitoring the impacts of pollution and other stressors on aquatic ecosystems.

### PROJECT FEATURES

The Water Quality Monitoring and Forecasting System is equipped with a range of features designed to enhance water quality management effectively. It includes real-time monitoring through strategically deployed sensors that continuously track essential parameters such as pH, turbidity, and contaminants. A user-friendly dashboard visualizes both real-time and historical data trends, allowing stakeholders to quickly assess water quality conditions. Advanced predictive analytics utilize historical data and machine learning algorithms to forecast potential water quality issues, enabling proactive risk mitigation. Automated alerts notify users when water quality parameters exceed safe thresholds, ensuring timely responses to hazards.

# SYSTEM ANALYSIS

## 2.SYSTEM ANALYSIS

### SYSTEM ANALYSIS

The Water Quality Monitoring and Forecasting System (WQMFS) is designed to continuously monitor key water quality parameters and utilize machine learning to analyze data and forecast potential issues, ensuring safe water for consumption and environmental sustainability. Key stakeholders include environmental agencies, water resource managers, local governments, and research institutions. The system's functional requirements involve real-time data collection via sensors measuring parameters like pH and turbidity, centralized data management for preprocessing and storage, and the implementation of machine learning algorithms for anomaly detection and predictive modeling. A user-friendly web-based dashboard will visualize data and generate automated reports for regulatory compliance. Non- functional requirements emphasize performance, scalability, security, and reliability. The system architecture comprises a data acquisition layer for sensor input, a processing layer for analysis, and a user interface layer for stakeholder access. Data flows from sensors to the central database, where it is processed and made available on the dashboard. Challenges include ensuring sensor calibration, maintaining data quality, and training users to effectively interpret results. Evaluation metrics will focus on the accuracy of predictions, system uptime, user engagement, and incident response time, enabling a proactive approach to water quality management.

### PROBLEM DEFINITION

Water Quality Monitoring and Forecasting System project centers around the growing concerns regarding water quality and its impact on public health and the environment. Many communities face challenges such as pollution, contamination, and inadequate monitoring of water bodies, which can lead to unsafe drinking water and detrimental effects on aquatic ecosystems.

Existing monitoring systems may lack real-time data collection capabilities, making it difficult to promptly identify and respond to water quality issues. Access to safe and clean water is a fundamental necessity for public health and environmental sustainability. However, many water sources face contamination from various pollutants, leading to adverse effects on human health and ecosystems. Traditional water quality monitoring methods are often insufficient due to their reliance on infrequent manual sampling, which fails to provide

timely data on changing conditions. This results in delays in detecting water quality issues, impeding effective response and management strategies.

The lack of real-time monitoring and predictive capabilities makes it challenging for stakeholders, such as environmental agencies and water resource managers, to respond proactively to potential contamination events. Additionally, existing systems may not effectively integrate data from multiple sources, hindering comprehensive analysis and decision-making. As a result, there is a pressing need for an advanced Water Quality Monitoring and Forecasting System that can provide continuous, real-time data collection and analysis, leveraging machine learning to predict future water quality trends and anomalies.

### EXISTING SYSTEM

Water quality monitoring (WQM) sensor technology has improved in recent years. Sensorized equipment that can autonomously measure the important physical, chemical, and biological characteristics is now easily available and is being deployed on buoys, boats, and ships at a low cost. Due to a lack of standardized methodologies for data collection and processing, spatio temporal volatility of critical parameters in water bodies, and novel contaminants, there is a gap between data quality, data gathering, and data interpretation. These gaps can be filled by deploying a network of multi parametric sensor systems in water bodies and using autonomous vehicles like marine robots and aerial vehicles to expand data coverage in space and time. For standardized data analysis and forecasting, sophisticated algorithms could also be used.

### LIMITATIONS OF EXISTING SYSTEM

* Less accuracy
* Low Efficiency

### PROPOSED SYSTEM

The proposed method is to build a machine learning model for Water quality. The process carries from data collection where the past data related to Water qualities are collected. Data mining is a commonly used technique for processing enormous data in the domain. The water if found before proper treatment can save lives. Machine learning is now applied and mostly used in health care where it reduces the manual effort and a better model makes error less which leads to saving the life. The data analysis is done on the dataset proper variable identification is done that is both the dependent variables and independent variables are found. Then proper machine learning algorithms are applied to the dataset where the pattern of data is learned. After applying different algorithms a better algorithm is used for the prediction of the outcome.

### ADVANTAGES OF THE PROPOSED SYSTEM

* High accuracy
* High efficiency

### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

* Economical Feasibility
* Technical Feasibility
* Social Feasibility

### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### HARDWARE & SOFTWARE REQUIREMENTS

* + 1. **HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

* System : i5.
* Ram : 4 GB.
* Hard Disk : 40 GB
  + 1. **SOFTWARE REQUIREMENTS:**

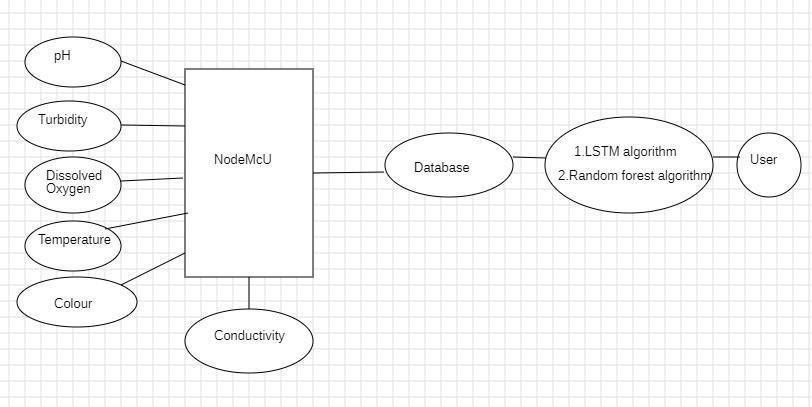
Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

* Operating system : Windows8.
* Coding Language : python 3.7.0

# 3. ARCHITECTURE

## 3. ARCHITECTURE

* 1. **PROJECT ARCITECTURE**



**3.1 Project Architecture**

**DESCRIPTION**

**Input Data:** Input data typically consists of measurements collected from various sensors and environmental sources. Common features might include:

* pH Level
* Dissolved Oxygen (DO)
* Turbidity
* Colour
* Temperature
* Conductivity

**Node MCU (central block):** The input data is being collected and processed by a Node MCU.

**Database :** The data processed by the Node MCU is sent to a database.

**Algorithms :** The data in the database is analyzed using machine learning algorithms, specifically:

* LSTM algorithm(Long Short-Term Memory, typically used for time series prediction).
* Random Forest algorithm (used for classification and regression tasks).

**User :** The user interacts with the system, likely receiving the results from the algorithms for decision-making or further analysis.

### USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

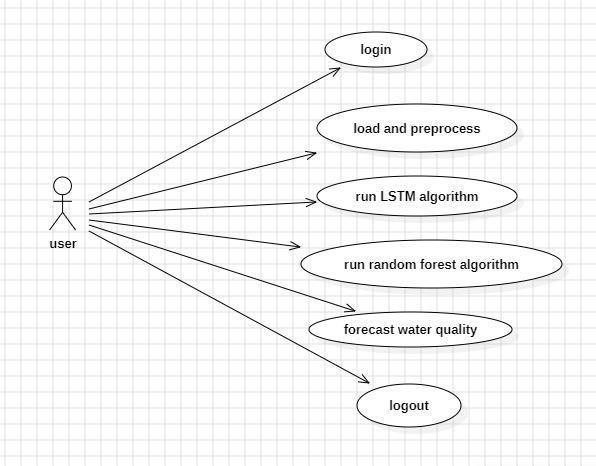


Figure 3.2: Use Case Diagram for user for Water Quality Monitoring And Forecasting System

### DESCRIPTION

Use case diagram represents illustrating the interactions between a user and the system's functionalities. The person icon represents the user, who interacts with the system, likely through a user interface. Each arrow signifies the actions the user can perform or trigger in the system. These arrows point to different system functionalities or processes.The ovals represent the main actions or processes the system provides for the user. The user can perform these actions in the sequence. The user first logs into the system. This step ensures that only authorized users can access the functionalities. After logging in, the system loads the data (possibly from sensors or a database) and preprocesses it for further analysis. Preprocessing can involve cleaning the data, normalizing values, handling missing data, etc. The user can run the LSTM (Long Short-Term Memory) algorithm, which is a machine learning model typically used for predicting time-series data, such as water quality trends over time. The user can also choose to run the \*Random Forest\* algorithm, which is used for classification or regression tasks. This might help in making decisions about water quality based on various sensor readings. After running the algorithms, the system forecasts the water quality, which provides predictions or insights based on the analyzed data .Finally, once the user is done using the system, they can log out, which ends the session and secures the system from unauthorized access.

### CLASS DIAGRAM

A class diagram is a static structure diagram used in object-oriented design to represent the classes, their attributes, methods, and the relationships among objects in a system. It typically consists of rectangles divided into sections, where the top section displays the class name, the middle section lists attributes, and the bottom section outlines methods. Relationships are illustrated through various types of connections, such as association, aggregation, composition, and inheritance, each depicting different kinds of interactions between classes. Class diagrams serve multiple purposes, including visualizing the system’s architecture, providing a blueprint for development, and facilitating communication among stakeholders. To create effective class diagrams, it’s important to keep them simple, use clear naming conventions, and organize related classes cohesively. Overall, class diagrams are essential tools that enhance understanding and clarity in software development.

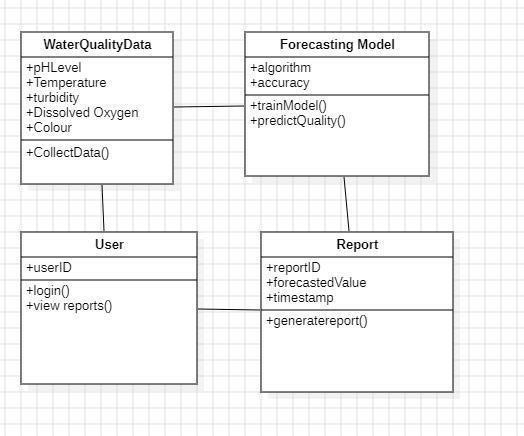


Figure 3.3: Class Diagram for Water Quality Monitoring And Forecasting System

### DESCRIPTION

Class diagram gives the relationships between different components of a water quality monitoring system. It includes four classes Water Quality Data, Forecasting Model, User, and Report, each with their own attributes and methods.

1. Water Quality Data: This class captures the data collected from water quality sensors. Attributes include pH Level, Temperature, Turbidity, Dissolved Oxygen, and colour.These represent different environmental parameters measured by sensors.The method CollectData() is responsible for gathering the sensor data.
2. Forecasting Model :This class focuses on the predictive models used to analyze the water quality data. Attributes include Algorithm and accuracy, which represent the algorithm used (e.g., LSTM or Random Forest) and the accuracy of the model. Methods include: trainModel(): Trains the model on water quality data.

predictQuality(): Predicts the water quality based on the trained model.

1. User:This class defines user-related attributes and actions. Attributes include user ID which uniquely identifies a user.Methods include login() Allows users to log into the system view

reports() allows users to view previously generated reports.

1. Report:This class generates reports based on predictions made by the forecasting model. Attributes include:Report ID (unique identifier for the report), forecastedValue (predicted water quality), and timestamp (when the report was generated).The method generatereport() creates and stores reports.

### SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

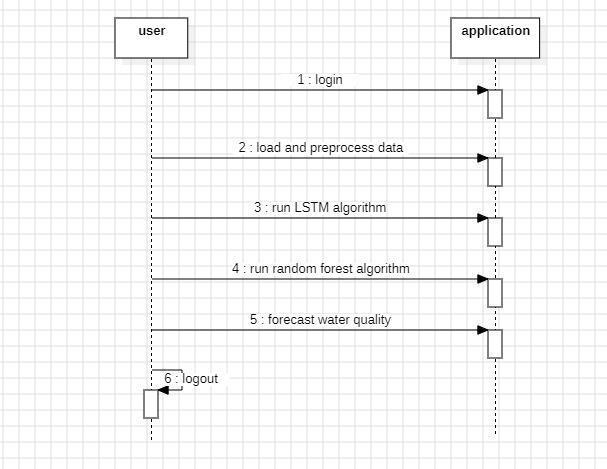


Figure 3.4: Sequence Diagram for Water Quality Monitoring And Forecasting System

### DESCRIPTION

A sequence diagram showing the interaction between a user and an application. Here's a brief explanation of each step in the diagram:

1. Login: The user logs into the application.
2. Load and preprocess data: The application loads the data and preprocesses it for further analysis.
3. Run LSTM algorithm: The application runs the Long Short-Term Memory (LSTM) algorithm, often used for time series forecasting or sequence prediction.
4. Run random forest algorithm: The application then runs a Random Forest algorithm, which is commonly used for classification or regression tasks.
5. Forecast water quality: Using the results from both algorithms, the application forecasts the water quality.
6. Logout: The user logs out of the application.

### ACTIVITY DIAGRAM

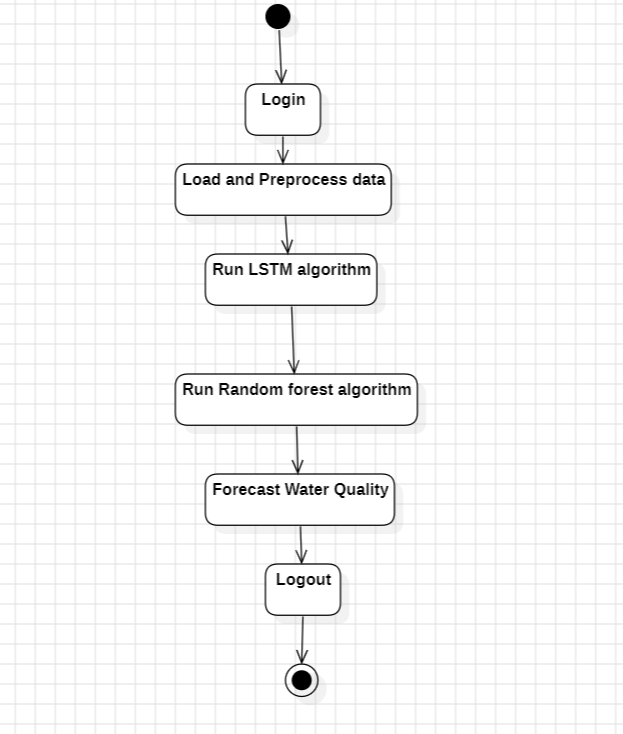
An activity diagram is a type of behavioral diagram in UML (Unified Modeling Language) that visually represents the flow of activities in a system or process. It is commonly used in software engineering to model the dynamic aspects of a system. 

Figure 3.5: Activity Diagram for Water Quality Monitoring And Forecasting System

### DESCRIPTION

1. Login: The user initiates a login process with the application.
2. Load and preprocess data: The application loads the necessary data and preprocesses it.
3. Run LSTM algorithm: The application runs the Long Short-Term Memory (LSTM) algorithm.
4. Run random forest algorithm: Next, the application runs the Random Forest algorithm.
5. Forecast water quality: The application provides a forecast of water quality using the outputs of the algorithms.
6. Logout: The user logs out of the application, completing the process.

# 4.IMPLEMENTATION

## IMPLEMENTATION

### LONG - SHORT TERM MEMORY

LSTM is a particular type of RNN with an extensive range of uses such as document classification, time series analysis, voice and speech recognition. Opposite to feed forward networks, the predictions (created by RNNs) are dependent on prior estimations. In experimental works, RNNs are not applied broadly due to include a few lacks that result in impractical estimations. Without investigation of too much detail, LSTM solves the problems by employing assigned gates for forgetting old information and learning new ones. The LSTM layer is made of our neural network layers that interact in a specific method. A usual LSTM unit involves three different parts, a cell, an output gate and a forget gate. The main task of the cell is recognizing values over random time intervals and the task of controlling the information flow into the cell and out of it belongs to the gates.

### RANDOM FOREST

Great number of decision trees make a random forest model. The model basically averages the forecast result of trees, which is named a forest. Also, the algorithm includes three random ideas, selecting training data randomly when forming trees, randomly choosing some subsets of variables when dividing nodes and deeming only a subset of all variables for splitting every node in each basic decision tree. Every basic tree learns from a random sample of the dataset during the training process of a random forest.

* + 1. **DATASET**

A dataset for a Water Quality Monitoring and Forecasting System is crucial for training machine learning models, conducting analyses, and making informed predictions about water quality.

We used data set having a well-structured is crucial. Below is an outline of what a dataset might look like for this task, including suggested features and an example of how the data could be organized. This is the URL of the dataset used in the project: <https://www.kaggle.com/datasets/ivivan/real-time-water-quality-data>

### 4.4 SAMPLE CODE

**Views.py**:

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import os

from django.core.files.storage import FileSystemStorage

import pymysql

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

import numpy as np

import pandas as pd

import keras

from keras import layers

from sklearn.model\_selection import train\_test\_split

from keras.utils.np\_utils import to\_categorical

import pickle

from sklearn.metrics import f1\_score

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from keras.layers import Dense, Dropout

import keras.layers

from sklearn.ensemble import RandomForestClassifier

global X, Y, dataset, X\_train, X\_test, y\_train, y\_test

global algorithms, accuracy, f1, precision, recall, classifier

def ProcessData(request):

if request.method == 'GET':

global X, Y, dataset, X\_train, X\_test, y\_train, y\_test

dataset = pd.read\_csv("Dataset/ml.csv")

dataset.fillna(0, inplace = True)

label = dataset.groupby('labels').size()

columns = dataset.columns

temp = dataset.values

dataset = dataset.values

X = dataset[:,2:dataset.shape[1]-1]

Y = dataset[:,dataset.shape[1]-1]

Y = Y.astype(int)

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

output += "<tr>"

for i in range(len(columns)):

output += "<th>"+font+columns[i]+"</th>"

output += "</tr>"

for i in range(len(temp)):

output += "<tr>"

for j in range(0,temp.shape[1]):

output += '<td><font size="" color="black">'+str(temp[i,j])+'</td>'

output += "</tr>"

context= {'data': output}

label.plot(kind="bar")

plt.title("Water Quality Graph, 0 (Good quality) & 1 (Poor Quality)")

plt.show()

return render(request, 'UserScreen.html', context)

def TrainRF(request):

global X, Y

global algorithms, accuracy, fscore, precision, recall, classifier

if request.method == 'GET':

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

cls = RandomForestClassifier()

cls.fit(X, Y)

classifier = cls

predict = cls.predict(X\_test)

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

a = accuracy\_score(y\_test,predict)\*100

algorithms.append("Random Forest")

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

arr = ['Algorithm Name', 'Accuracy', 'Precision', 'Recall', 'F1 Score']

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

output += "</tr>"

for i in range(len(algorithms)):

output +="<tr><td>"+font+str(algorithms[i])+"</td><td>"+font+str(accuracy[i])+"</td><td>"+font+str(precision[i])+"</td><td>"+font+str(recall[i])+"</td><td>"+font+str(fscore[i])+"</td></tr>"

context= {'data': output}

return render(request, 'UserScreen.html', context)

def TrainLSTM(request):

if request.method == 'GET':

global X, Y

global algorithms, accuracy, fscore, precision, recall

algorithms = []

accuracy = []

fscore = []

precision = []

recall = []

X1 = np.reshape(X, (X.shape[0], X.shape[1], 1))

Y1 = to\_categorical(Y)

print(X1.shape)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, Y1, test\_size=0.2)

if request.method == 'GET':

lstm\_model = Sequential()

lstm\_model.add(keras.layers.LSTM(100,input\_shape=(X\_train.shape[1], X\_train.shape[2])))

lstm\_model.add(Dropout(0.5))

lstm\_model.add(Dense(100, activation='relu'))

lstm\_model.add(Dense(y\_train.shape[1], activation='softmax'))

lstm\_model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

lstm\_model.fit(X1, Y1, epochs=40, batch\_size=32, validation\_data=(X\_test, y\_test))

print(lstm\_model.summary())#printing model summary

predict = lstm\_model.predict(X\_test)

predict = np.argmax(predict, axis=1)

testY = np.argmax(y\_test, axis=1)

p = precision\_score(testY, predict,average='macro') \* 100

r = recall\_score(testY, predict,average='macro') \* 100

f = f1\_score(testY, predict,average='macro') \* 100

a = accuracy\_score(testY,predict)\*100

algorithms.append("LSTM")

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

arr = ['Algorithm Name', 'Accuracy', 'Precision', 'Recall', 'F1 Score']

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

output += "</tr>"

for i in range(len(algorithms)):

output +="<tr><td>"+font+str(algorithms[i])+"</td><td>"+font+str(accuracy[i])+"</td><td>"+font+str(precision[i])+"</td><td>"+font+str(recall[i])+"</td><td>"+font+str(fscore[i])+"</td></tr>"

context= {'data': output}

return render(request, 'UserScreen.html', context)

def Predict(request):

if request.method == 'GET':

return render(request, 'Predict.html', {})

def PredictAction(request):

if request.method == 'POST':

global classifier

testFile = request.POST.get('t1', False)

test = pd.read\_csv("Dataset/testData.csv")

test.fillna(0, inplace = True)

test = test.values

X = test[:,2:dataset.shape[1]-1]

predict = classifier.predict(X)

print(predict)

arr = ['Test Data', 'Water Quality Forecasting Result']

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

output += "</tr>"

labels = ['Good Quality', 'Poor Quality']

for i in range(len(predict)):

output +="<tr><td>"+font+str(test[i])+"</td><td>"+font+str(labels[predict[i]])+"</td></tr>"

context= {'data': output}

return render(request, 'UserScreen.html', context)

def UserLogin(request):

if request.method == 'GET':

return render(request, 'UserLogin.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Signup(request):

if request.method == 'GET':

return render(request, 'Signup.html', {})

def UserLoginAction(request):

global uname

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

index = 0

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Waterquality',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username,password FROM signup")

rows = cur.fetchall()

for row in rows:

if row[0] == username and password == row[1]:

uname = username

index = 1

break

if index == 1:

context= {'data':'welcome '+uname}

return render(request, 'UserScreen.html', context)

else:

context= {'data':'login failed. Please retry'}

return render(request, 'UserLogin.html', context)

def SignupAction(request):

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

contact = request.POST.get('t3', False)

gender = request.POST.get('t4', False)

email = request.POST.get('t5', False)

address = request.POST.get('t6', False)

output = "none"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Waterquality',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username FROM signup")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

output = username+" Username already exists"

break

if output == 'none':

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Waterquality',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "INSERT INTO signup(username,password,contact\_no,gender,email,address) VALUES('"+username+"','"+password+"','"+contact+"','"+gender+"','"+email+"','"+address+"')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

output = 'Signup Process Completed'

context= {'data':output}

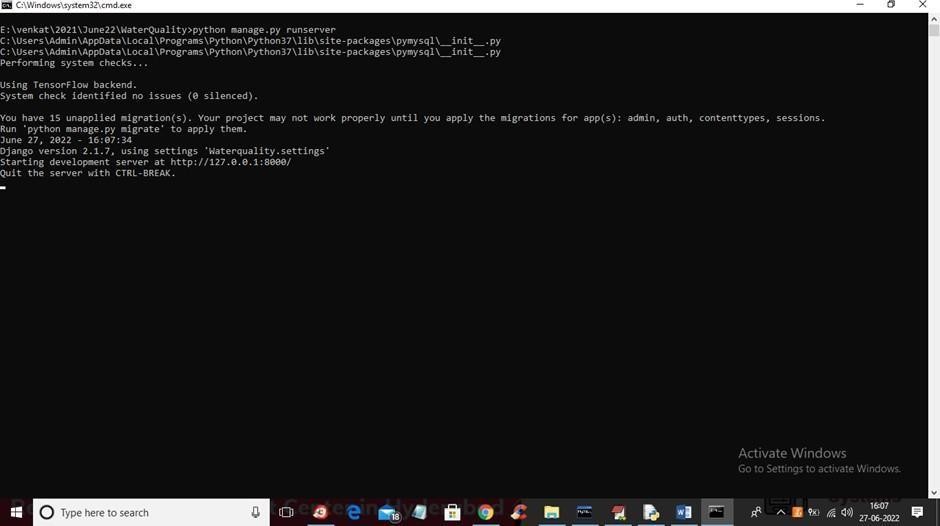
return render(request, 'Signup.html', context)

# SCREENSHOTS

### SCREENSHOTS

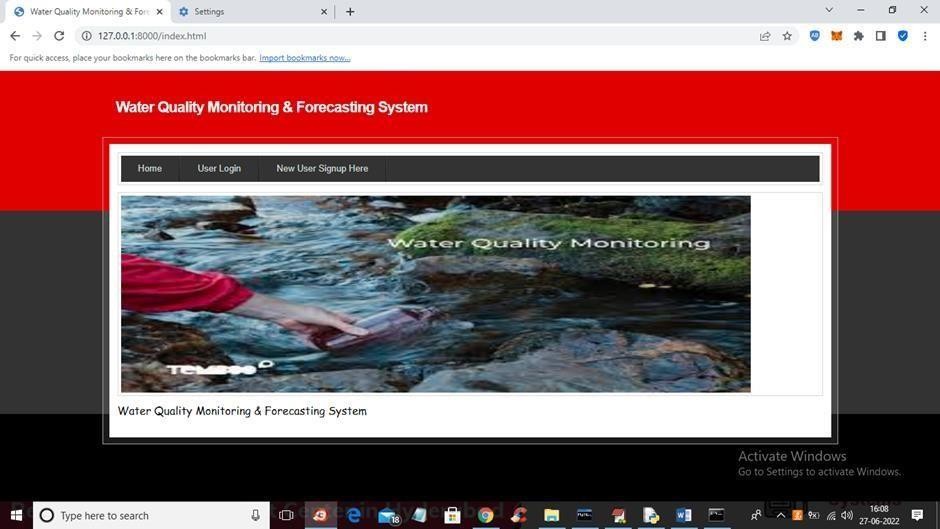
To run project first copy content from ‘DB.txt’ file and then paste in MYSQL database to create it and

now double click on ‘run.bat’ file to start DJANGO server and then will get below output



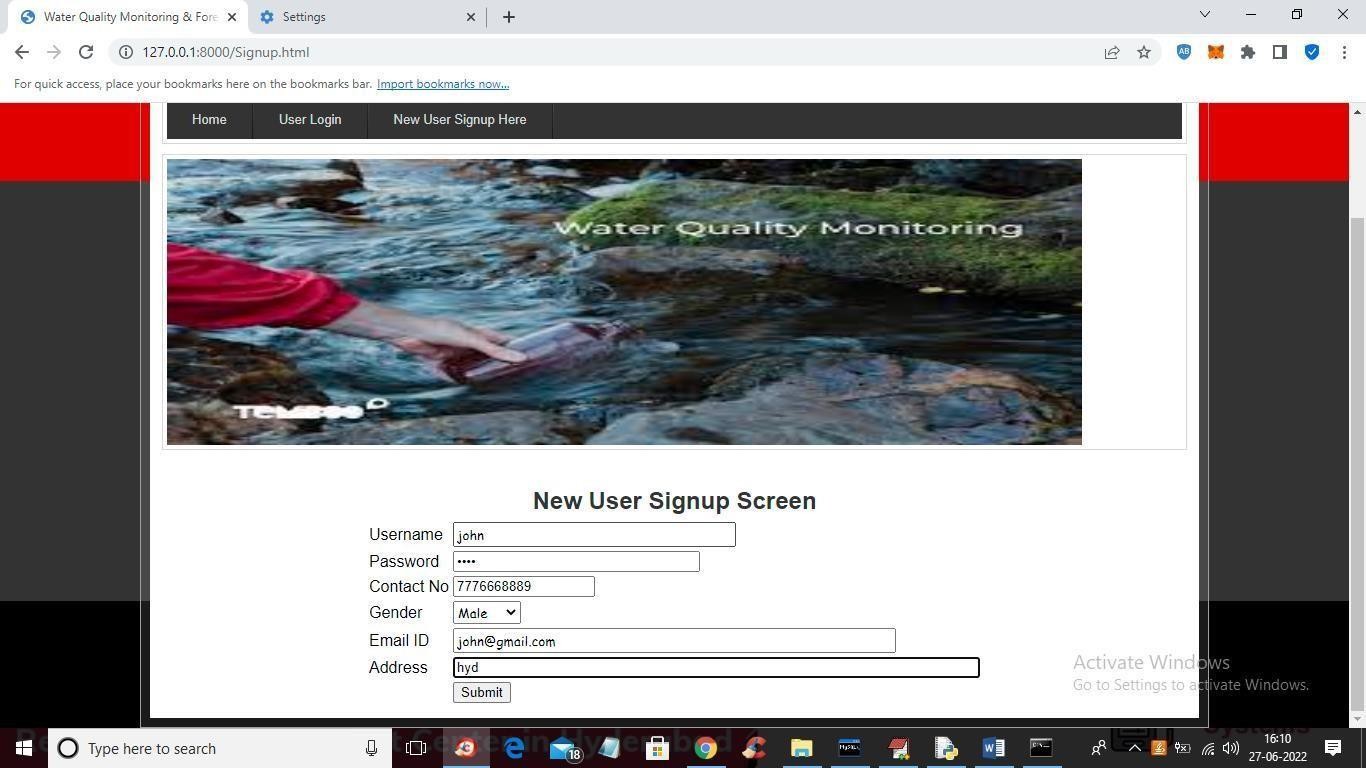
* + 1. Copy content from DB.txt file and paste in MYSQL

In above screen DJANGO server started and now open browser and enter URL as ‘http://127.0.0.1:8000/index.html’ and press enter key to get below page



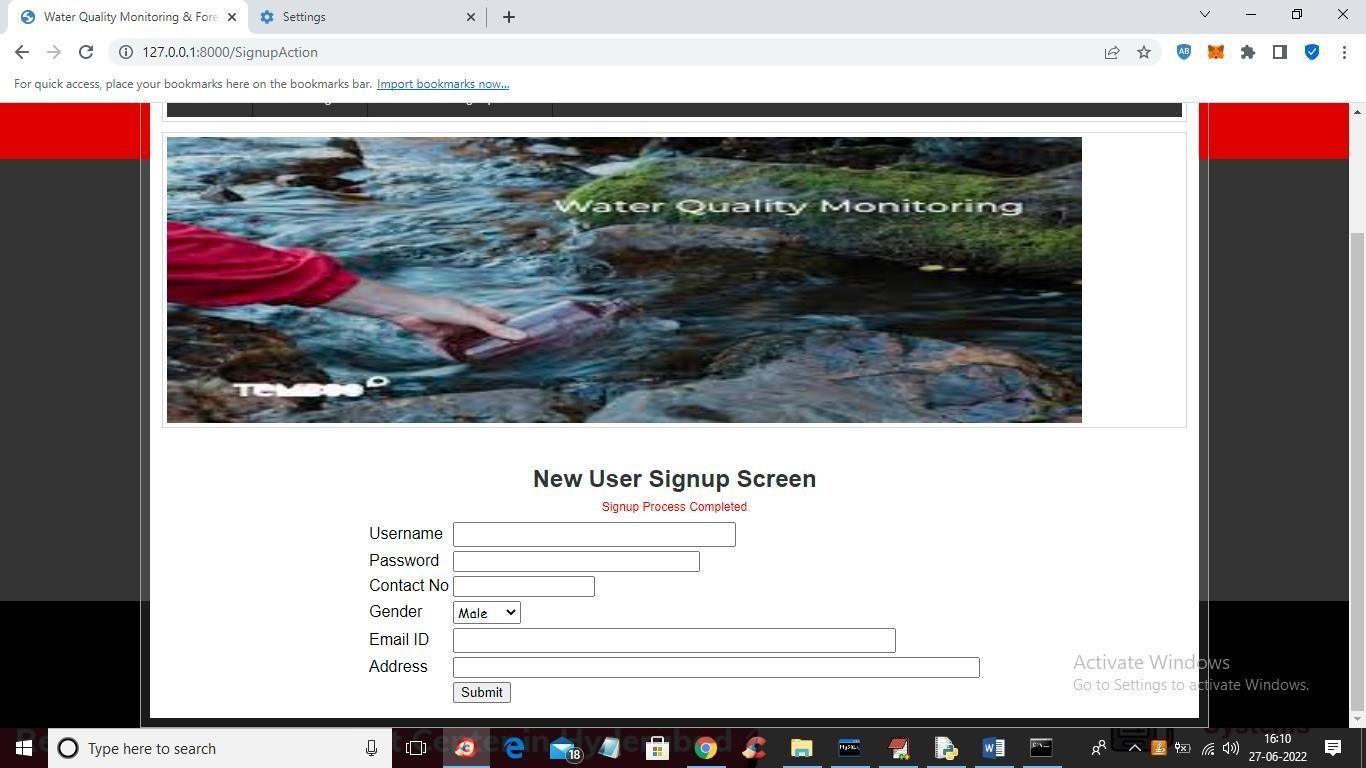
* + 1. Water Quality Monitoring and Forecasting System login page

In above screen click on ‘New User Signup Here’ link to get below screen



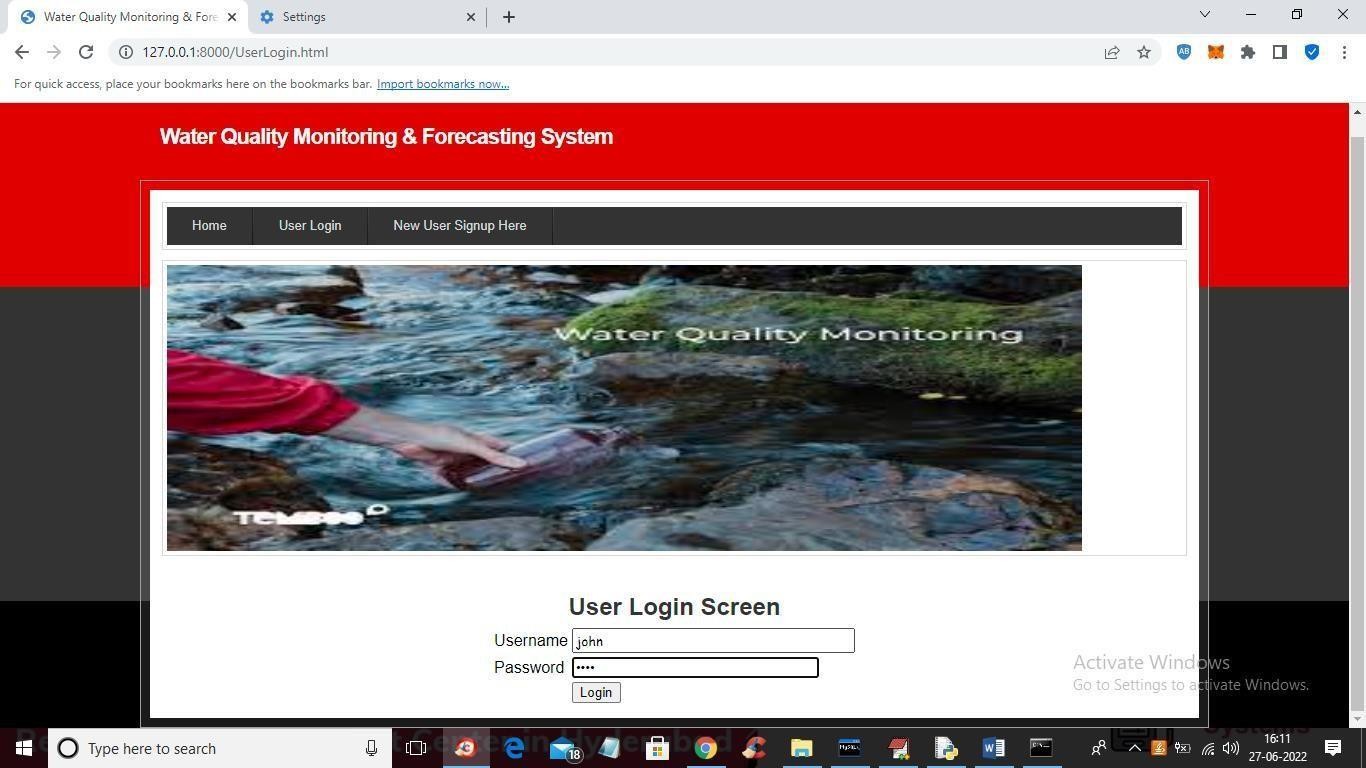
* + 1. New User Signup Page

In above screen user is signing up and then press button to get below screen



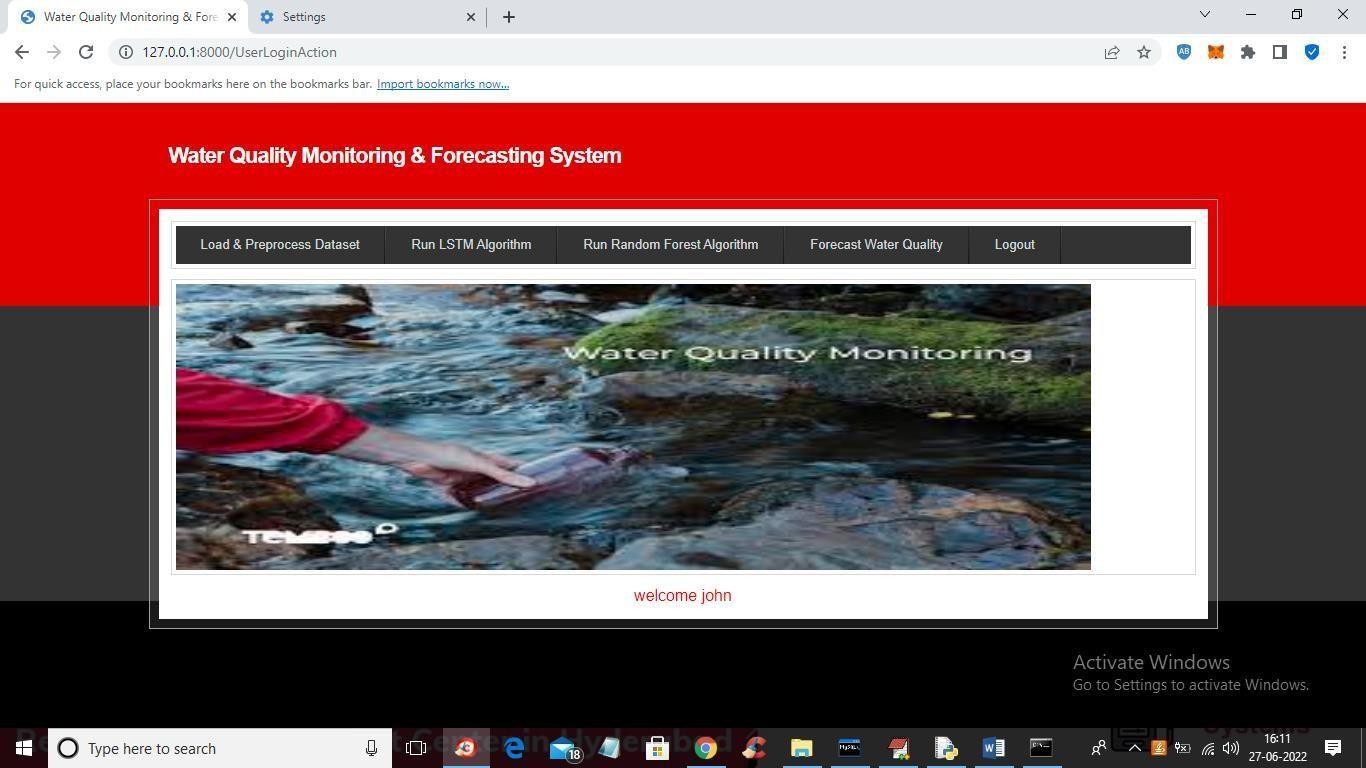
5.4 Signup process completed

In above screen signup process completed and now click on ‘User Login’ link to get below screen



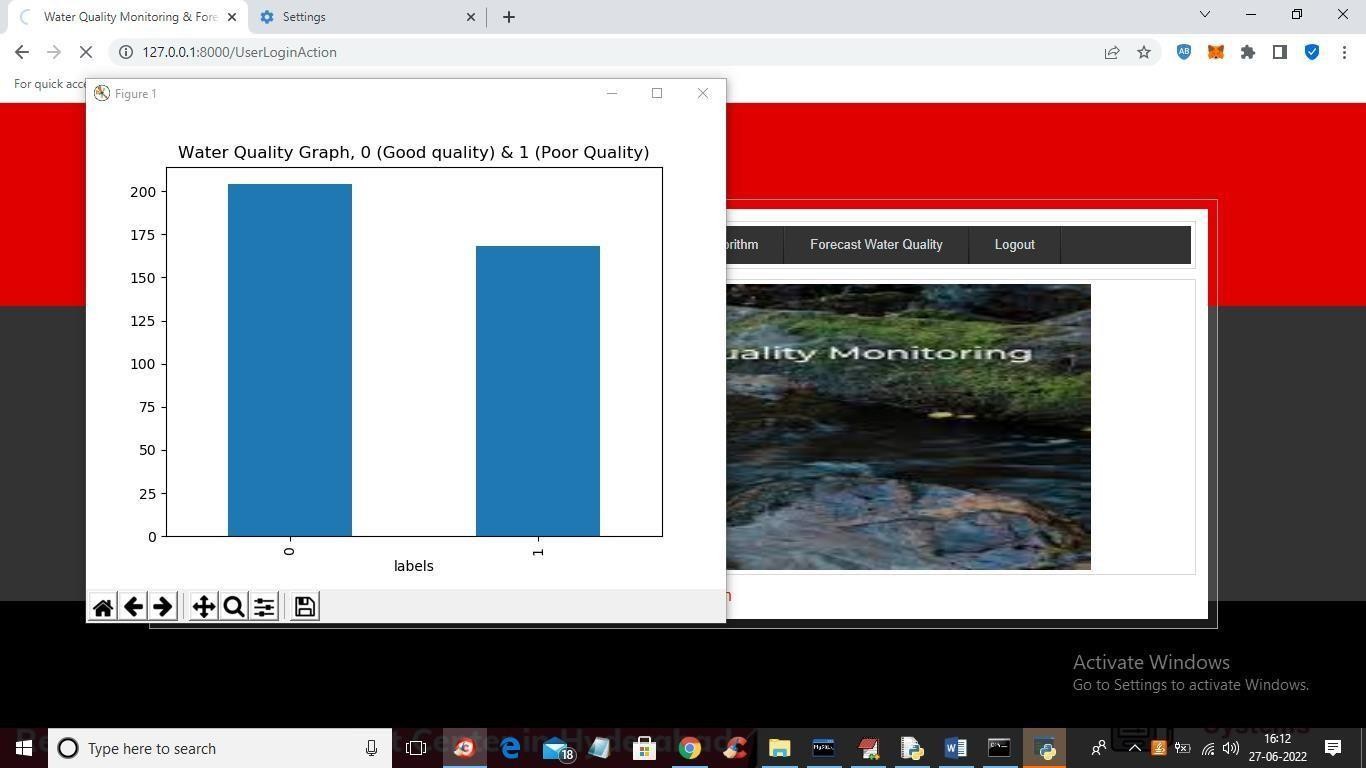
* + 1. User Login Screen

In above screen user login and after login will get below screen



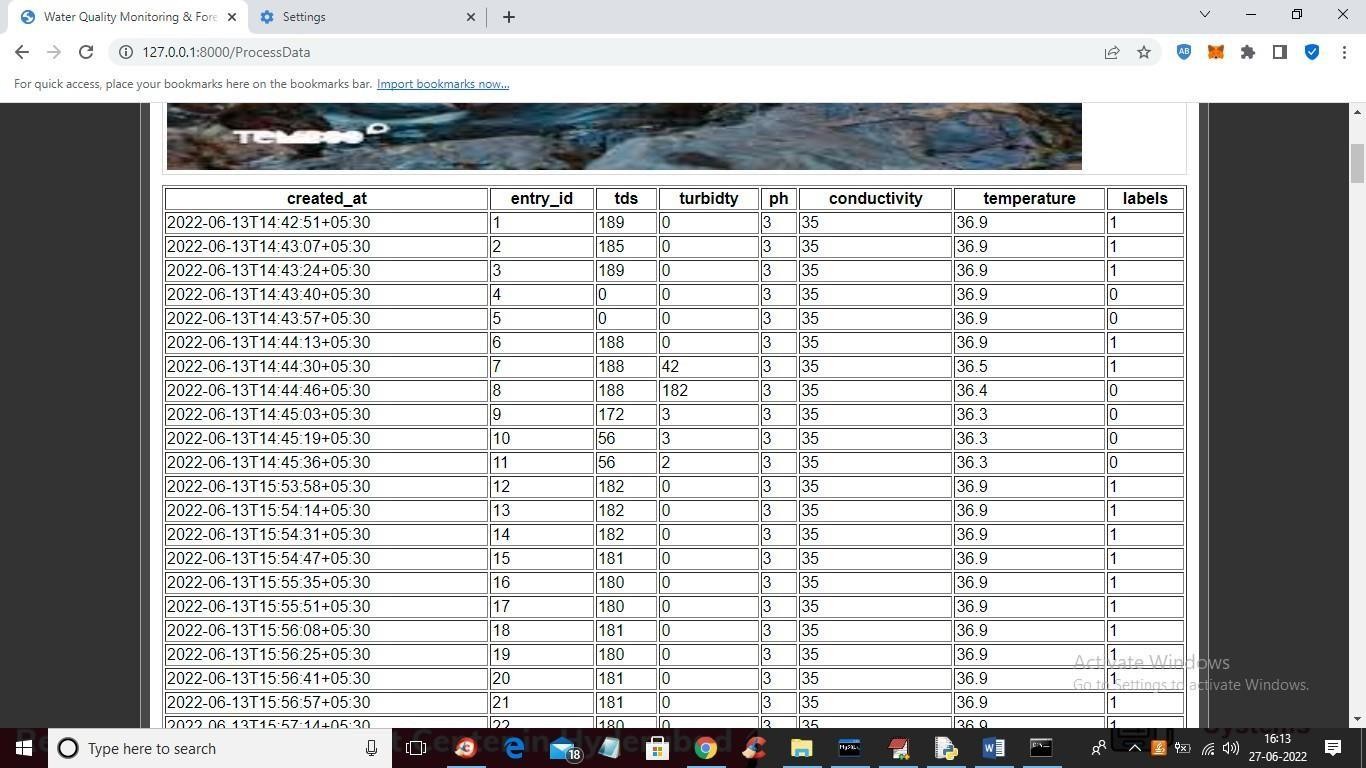
* + 1. Click on Load and Preprocess Data

In above screen click on ‘Load & Preprocess Dataset’ link to load and process dataset such as replacing missing values with 0 and then split dataset into train and test and get below output



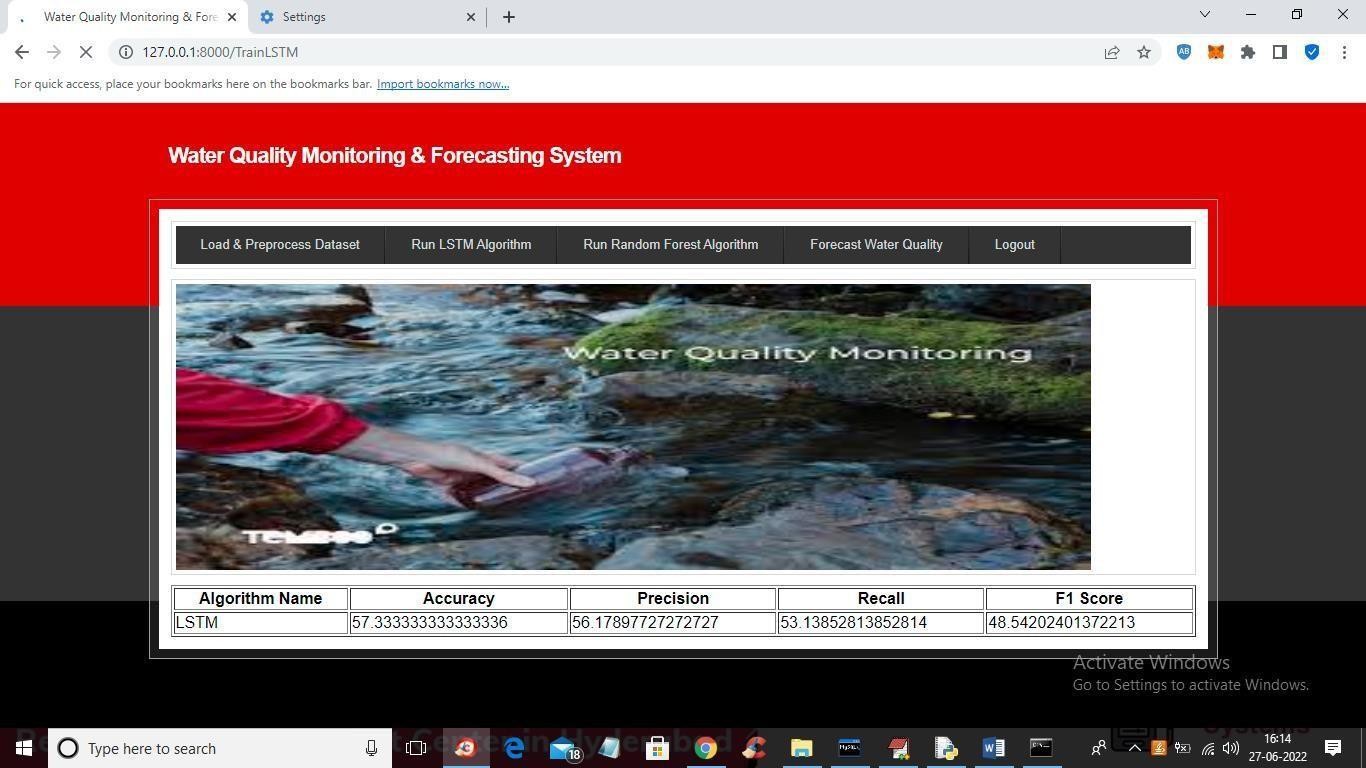
* + 1. Water Quality Graph

In above screen dataset is processed and in above graph x-axis contains water quality as 0 or 1 where 0 means GOOD quality and 1 means POOR quality and y-axis represents number of records and now close above graph to get below screen



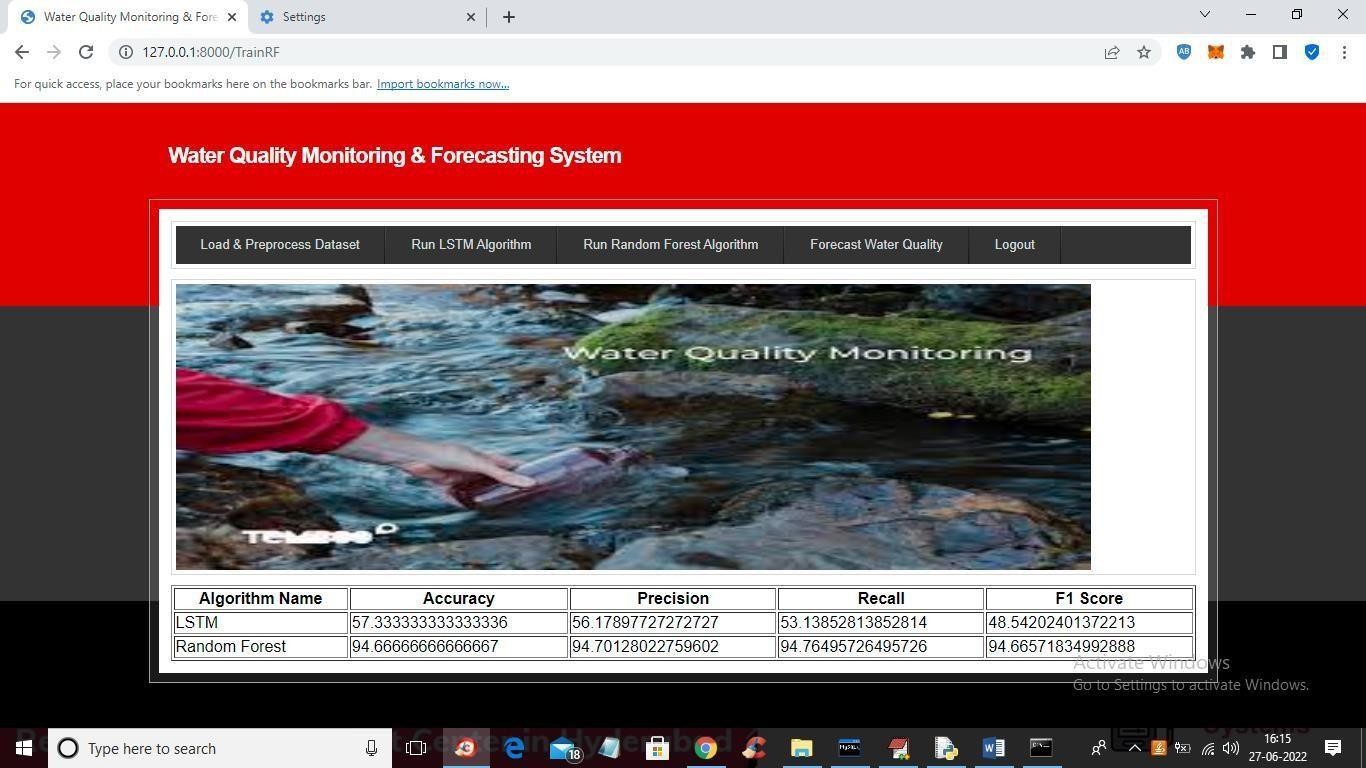
* + 1. Dataset processed and loaded

In above screen we can see dataset processed and loaded and now click on ‘Train LSTM Algorithm’ link to train LSTM and get below output



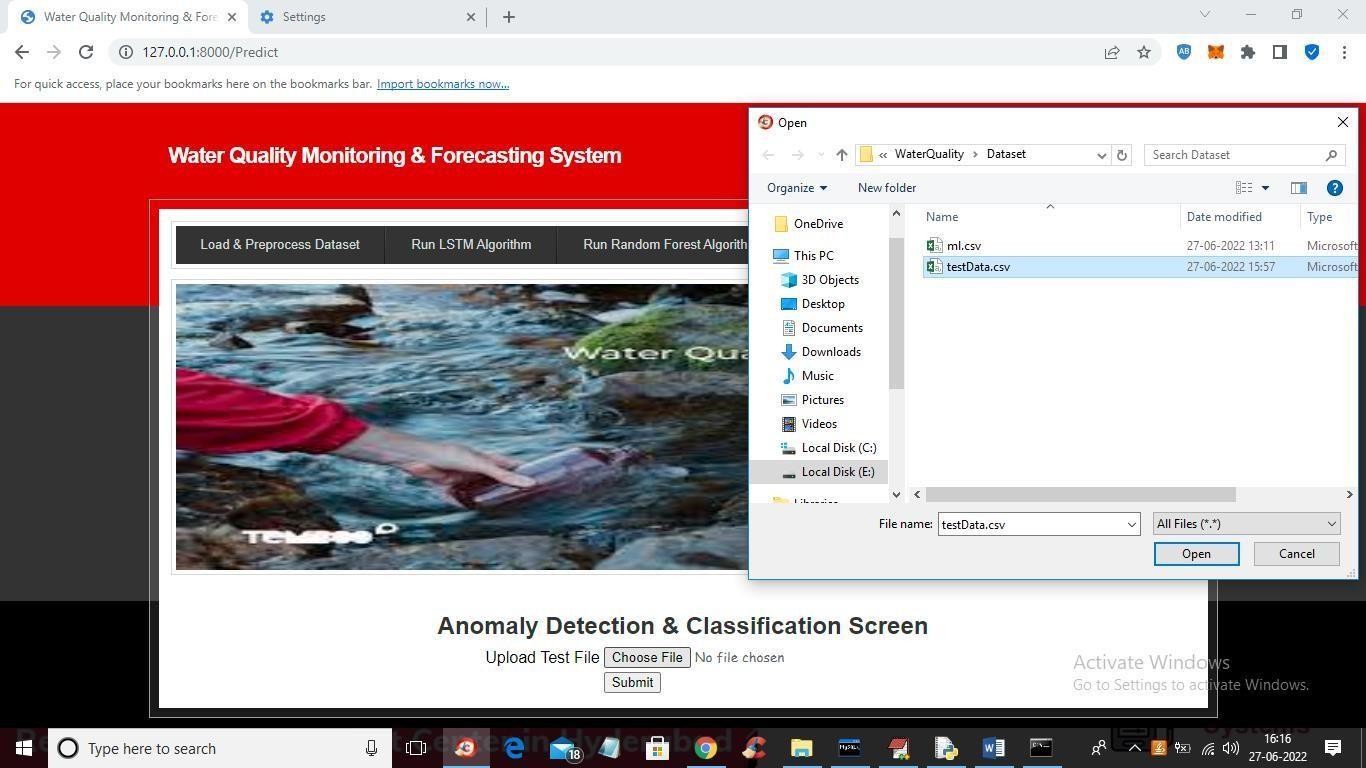
* + 1. LSTM Algorithm gives the output.

In above screen LSTM got trained and with LSTM we got 57% accuracy and now click on ‘Train Random Forest Algorithm’ link to train Random Forest and get below output



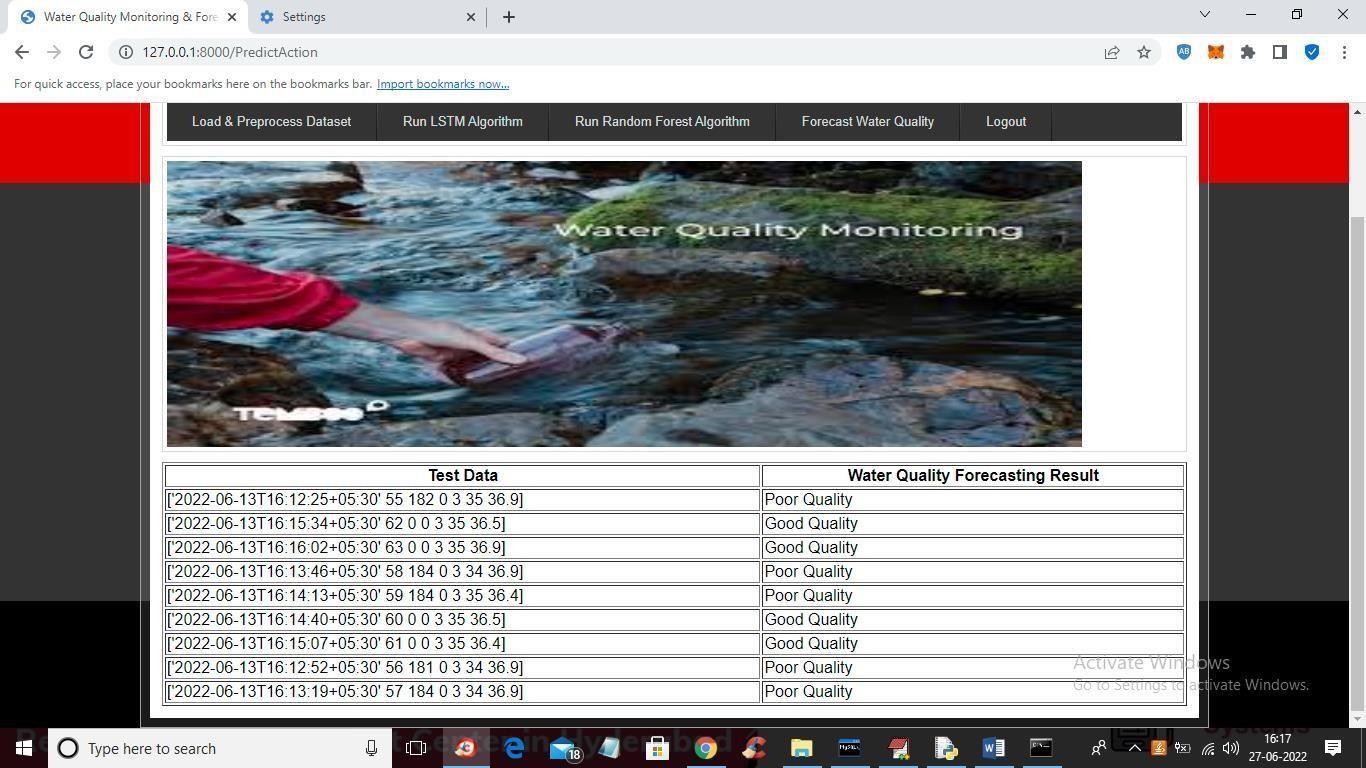
* + 1. Random Forest Algorithm output will be displayed.

In above screen with Random Forest we got 94% accuracy and now click on ‘Forecast Water Quality’ link to upload test data and then forecast quality



* + 1. Upload test data file

In above screen selecting and uploading ‘testData.csv’ file and then click on ‘Open’ and ‘Submit’ button to get below forecast output



* + 1. Water Quality Forecasting Result will be displayed.

In above screen in tabular output first column contains water test values and second column contains forecast result as ‘Poor’ or “Good”

# TESTING

## 6.TESTING

### INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discoverevery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTING

* + 1. **UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

|  |  |
| --- | --- |
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

* 1. **TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | Input | Expected Output | Actual Output | Test Case Pass/Fail |
| 1 | Signup Credentials | Username: Indhu Password: indhu18 Contact number:9876543210 Gender: Female Email id: [anu@gmail.com](mailto:anu@gmail.com) Address: Hyderabad | It should move to the user home page. | It should move to the user home page. | Pass |
| 2 | User Login | Username: Indhu Password: indhu18 | It should move to the home page. | It should move to  the user home page. | Pass |
| 3 | Dataset | Upload dataset | Dataset will be loaded | Dataset will be loaded. | Pass |
| 4 | Run LSTM  Algorithm and Random Forest Algorithm | Dataset will be processed | Accuracy, precision, Recall, F1 score will be displayed. | Accuracy, precision, Recall, F1 score will be displayed. | Pass |
| 5 | Water Quality  Forecasting Result | 0 or 1 | It displays 0 for  good quality and 1 for bad quality | It displays 0 for  good quality and 1 for bad quality | Pass |

# 7.CONCLUSION

**& FUTURE SCOPE**

## CONCLUSION & FUTURE SCOPE

### PROJECT CONCLUSION

Data cleaning and processing, missing value analysis, exploratory analysis, and model creation and evaluation were all part of the analytical process. The best accuracy on a public test set will be discovered, as will the highest accuracy score. This application can assist in determining the current state of water quality. The potability of the water, which is determined by factors like turbidity, pH, and TDS, will be properly predicted by our suggested method, indicating whether the water is suitable for human consumption or not. In this project, we offer a model that fuses artificial intelligence with the internet of things. Multiple sensors and microcontrollers are utilised to assess the TDS, pH, and turbidity data that were validated since the model used machine learning techniques. More sensors can be added in the future to expand the range of properties (such temperature, pH, and dissolved oxygen) that our model can examine.

### FUTURE SCOPE

The conductivity acts as a sensor gateway. The sensor input are sent to the pi4, a edge level processor(personal computer) where in the K Means, a machine learning algorithm is used for predicting the quality of water. The predicted water quality data are stored in Cloud server for future access. The predicted data is sent to the water controller unit for further action. This has brought about complete computerized Water Quality Monitoring framework utilizing IoT and AI Technologies by which the gadgets impart among themselves in anticipating the Water Quality for private country region. Subsequently, the water quality can be watched consequently with no human interference. The proposed system can be extended further by water retreatment mechanism.

# 8.BIBLIOGRAPHY

## BIBLIOGRAPHY

### 8.1 REFERENCES

1. M. Simić, G. M. Stojanović, L. Manjakkal, and K. Zaraska, "Multisensor system for remote environmental (air and water) quality monitoring,"
2. S. C. Mukhopadhyay and A. Mason, Smart sensors for real-time water quality monitoring. Springer, 2013.
3. Cloete N A,Malekian R,Nair L.Design of smart sensors for real-time water quality monitoring[J].IEEE access,2016,4:3975-3990.
4. M. Simić, G. M. Stojanović, L. Manjakkal, and K. Zaraska, "Multisensor system for remote environmental (air and water) quality monitoring," in 24th IEEE Telecomm. forum, 2016, pp. 1-4
5. D. Pooja, P. Kumar, P. Singh, and S. Patil, Sensors in Water Pollutants Monitoring: Role of Material. Springer, 2020.

**8.2 WEBSITE**

[**https://mahesh8075.github.io/Water-quality/**](https://mahesh8075.github.io/Water-quality/)