EFFICIENT WATER QUALITY ANALYSIS AND PREDICTION USING MACHINE LEARNING

Submitted by

JEYAKALEESWARI M (923819104016)

HARINEE K (923819104013)

KRISHNAPRIYA K (923819104022)

SUBANI S B (923819104045)

RISHWANA MUMTAJ S (923819104038)

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

MANGAYARKARASI COLLEGE OF ENGINEERING,
MADURAI 625 402

ANNA UNIVERSITY:: CHENNAI 600 025

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Project Review

Water is considered as a vital resource that affects various aspects of human health and lives. The quality of water is a major concern for people living in the urban areas. The quality of water serves as a powerful environmental determinant and a foundation for the prevention and control of waterborne diseases. However predicting the urban water quality is a challenging task since the water quality varies in urban spaces non-linearly and depends on multiple factors, such as meteorology, water usage patterns, and land uses, so this project aims at building a Machine Learning (ML) model to Predict Water Quality by considering all water quality standard indicators. The proposed methodology achieves reasonable accuracy using a minimal number of parameters to validate the possibility of its use in real time water quality detection systems.

1.2 Purpose

Water is an inorganic, transparent, and colourless chemical substance that is required for the survival of most existing organisms and humans. Water of sufficient quality is required for living creatures to survive. Water species can only withstand a certain amount of pollution. Exceeding these boundaries has an impact on the existence of these organisms and puts their lives in jeopardy. Water quality monitoring is a must to keep a reliable and safe water supply. Water contamination has become increasingly significant as the economy has grown and urbanization has expanded. Predicting water efficiency factors is a troublesome element of any hydrophyte system’s examination. The study of various strategies for predicting water quality in reservoirs has both theoretical and practical implications . Several variables can affect water efficiency, including the calcium content in natural water, as well as industrial waste, stones, and contaminated sewage, which can all be considered calcium sources, resulting in poor water quality. Moreover, hard water is linked to heart disease and a slew of other significant and chronic illnesses. The impurity of water sources by chemical, physical, and radioactive pollutants is the most common cause of water pollution .

CHAPTER-2

LITERATURE SURVEY

2.1 Existing problem

The alarming consequences of poor water quality necessitate an alternative method, which is quicker and inexpensive. With this motivation, this research explores a series of supervised machine learning algorithms to estimate the water quality index (WQI), which is a singular index to describe the general quality of water, and the water quality class (WQC), which is a distinctive class defined on the basis of the WQI. The Existing methodology employs four input parameters, namely, temperature, turbidity, pH and total dissolved solids. Of all the employed algorithms, gradient boosting, with a learning rate of 0.1 and polynomial regression, with a degree of 2, predict the WQI most efficiently, having a mean absolute error (MAE) of 1.9642 and 2.7273, respectively. Whereas multi-layer perceptron (MLP), with a configuration of (3, 7), classifies the WQC most efficiently, with an accuracy of 0.8507. The proposed methodology achieves reasonable accuracy using a minimal number of parameters to validate the possibility of its use in real time water quality detection systems.

2.2 References

1. PCRWR. National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006); Pakistan Council of Research in Water Resources Islamabad: Islamabad, Pakistan, 2007. Available online: http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/Water%20Quality%20Monitoring%20Report%202005-06.pdf (accessed on 23 August 2019).

2. Mehmood, S.; Ahmad, A.; Ahmed, A.; Khalid, N.; Javed, T. Drinking Water Quality in Capital City of Pakistan. Open Access Sci. Rep. 2013, 2. [CrossRef]

3. PCRWR. Water Quality of Filtration Plants, Monitoring Report; PCRWR: Islamabad, Pakistan, 2010.Availableonline:http://www.pcrwr.gov.pk/Publications/Water%20Quality%20Reports/FILTRTAION%20PLANTS% 20REPOT-CDA.pdf (accessed on 23 August 2019).

4. Gazzaz, N.M.; Yusoff, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F. Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors. Mar. Pollut. Bull. 2012, 64, 2409–2420. [CrossRef]

2.3 Problem statement

"Efficient water quality analysis and prediction using machine learning"

For the WQI prediction, artificial neural network models, namely nonlinear autoregressive neural network (NARNET)and long short-term memory (LSTM) deep learning algorithm, have been developed. In addition, three machine learning algorithms, namely Decision tree,K-nearest neighbour (K-NN), and Naive Bayes, have been used for the WQC forecasting. The used dataset has 9 significant parameters, and the developed models were evaluated based on some statistical data.

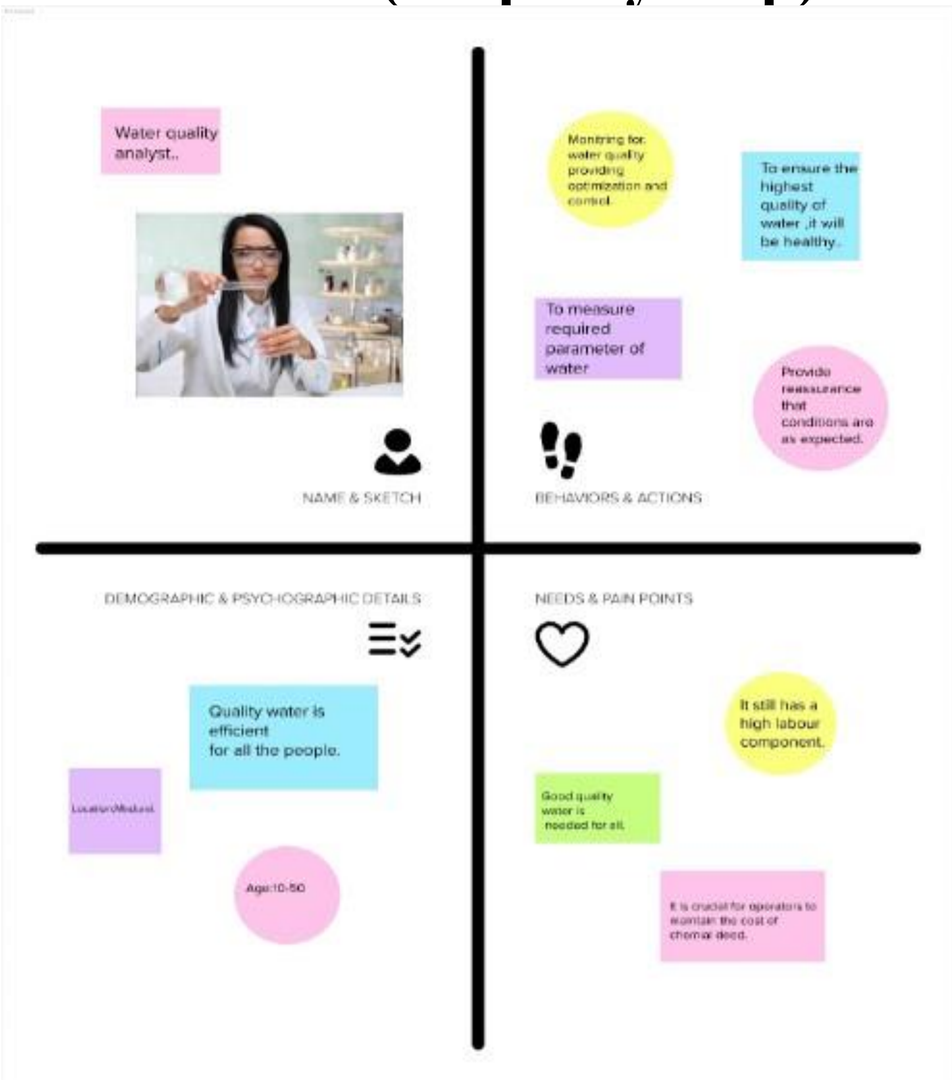
CHAPTER-3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.

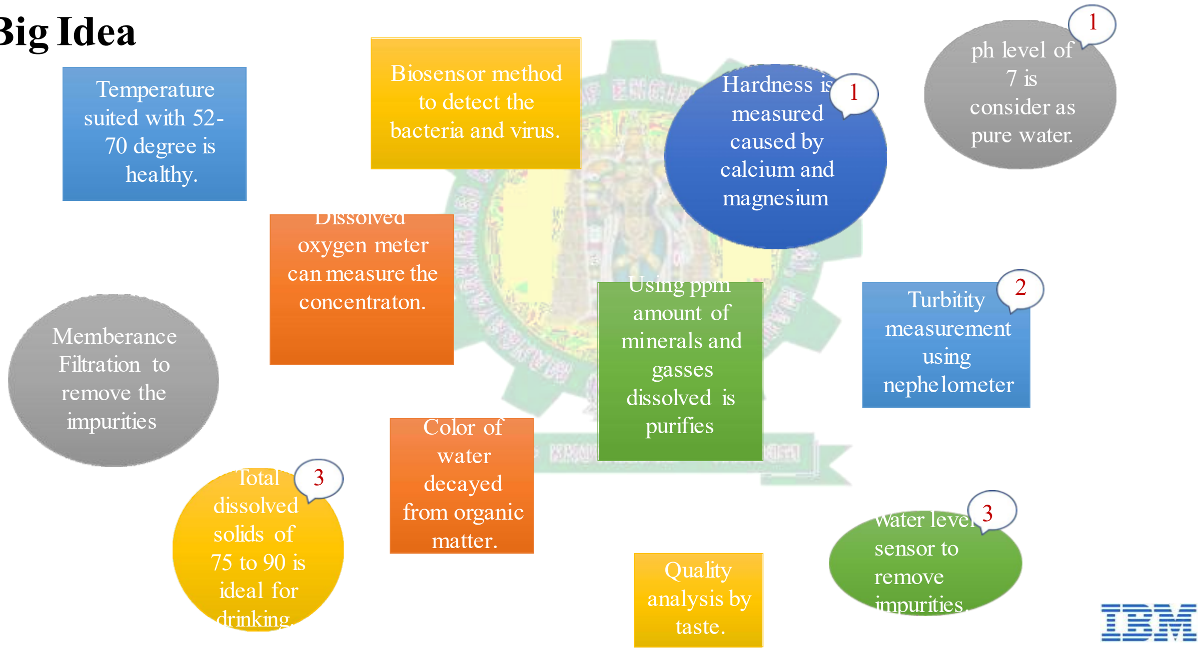
Persona & Context(Empathy map)



3.2 Ideation and Brainstorming

It consists of all the ideas of instruments and equipments that we are going to implement in this project

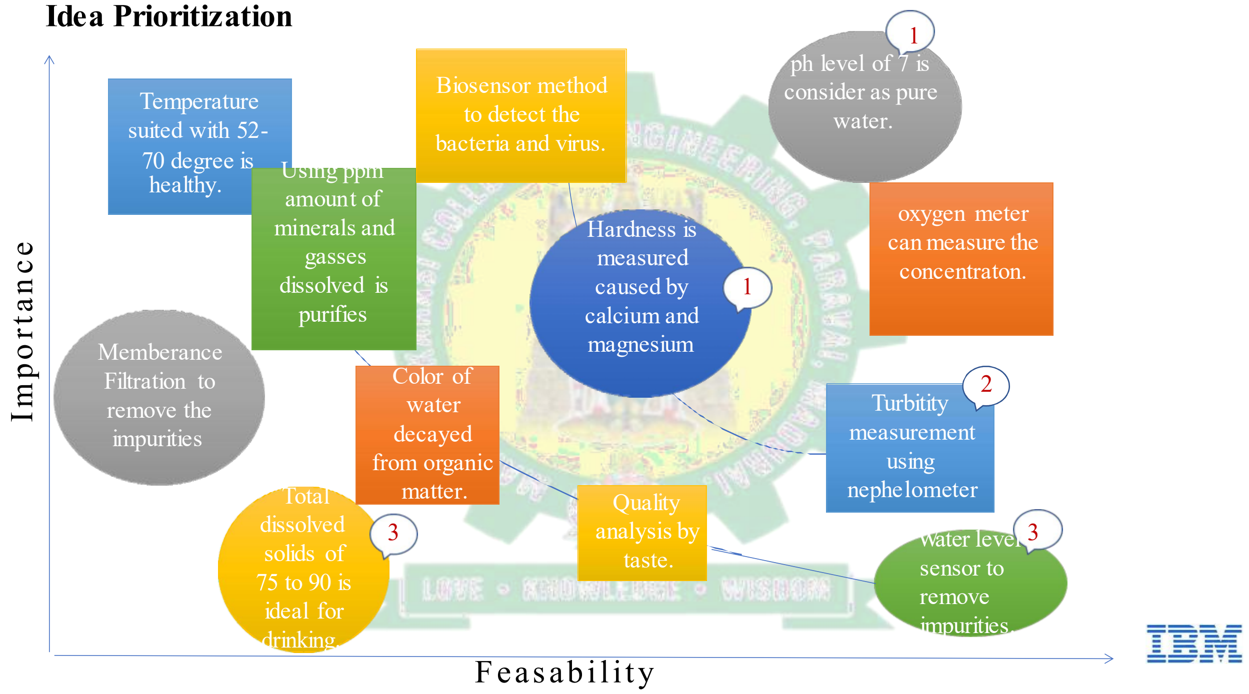
Big Idea



Prioritization:

It deals with the prioritizing of the big ideas in order of highest to lowest likes.

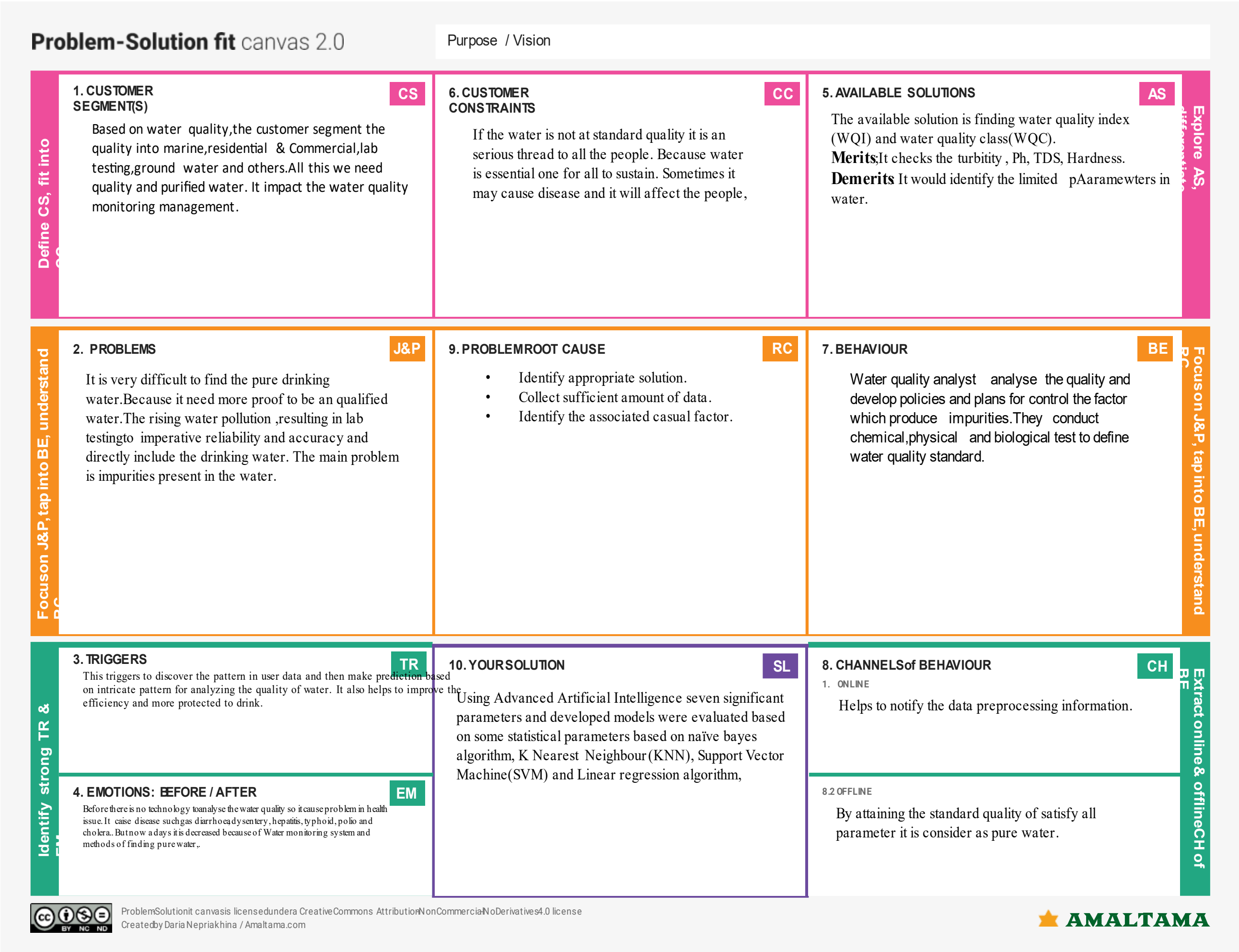
Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Efficient Water Quality Analysis and Prediction using Machine Learning.
2.	**Idea / Solution description**	For the WQI prediction, artificial neural network models, namely nonlinear autoregressive neural network (NARNET) and long short-term memory (LSTM) deep learning algorithm, have been developed. In addition, three machine learning algorithms, namely, support vector machine (SVM), K-nearest neighbour (K-NN), and Naive Bayes, have been used for the WQC forecasting. The used dataset has 7 significant parameters, and the developed models were evaluated based on some statistical parameters
3.	**Novelty / Uniqueness**	In previous they find water quality with help of WQI and WQC. Now the solution is find with help of advanced artificial intelligence and it include seven parameters
4.	**Social Impact / Customer Satisfaction**	During the last years, water quality has been threatened by various pollutants. Therefore, modelling and predicting water quality have become very important in controlling water pollution. In this work, advanced artificial intelligence (AI) algorithms are developed to predict water quality index (WQI) and water quality classification (WQC).This is the impact of this statement.
5.	**Business Model (Revenue Model)**	*The revenue stream include the Promoted trends and method. Technology and production is improved in business side. It increased the profit and also the logistic way.*
6.	**Scalability of the Solution**	Scalability of this solution can handle any amount of data and perform many computations in a cost effective and time saving to instantly serve millions of users residing at global location.

3.4 Problem Solution Fit



CHAPTER-4

Requirement Analysis

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registration	Registration through Gmail Create an account Follow the instructions
FR-2	User Confirmation	Confirmation via Email and it is predicted by water level sensor
FR-3	Interface sensor	Interface sensor and Water level sensor produces the detection of clean drinking water
FR-4	Accessing datasets	Datasets are collected by data preprocessing method.
FR-5	Mobile application	The efficient of water quality is analyzed, the mobile application is not used .

4.2 Non-Functional Requirement

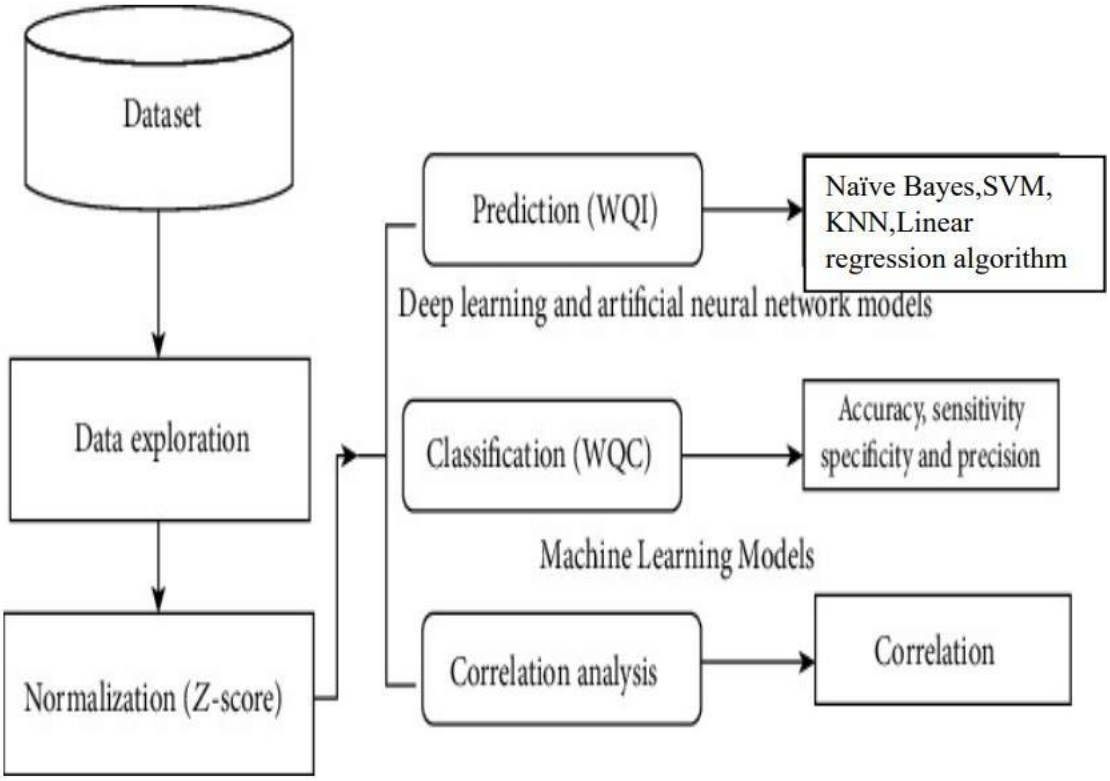
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This project is useful for all human being by predicting a purified water.
NFR-2	**Security**	We have designed this project to secure the people from drinking the impurity water.
NFR-3	**Reliability**	This project will help everyone in protecting their health. Accurate water quality prediction is the basis of water environment management and is of great significance for water environment protection.
NFR-4	**Performance**	This system uses different sensors for monitoring the water quality by determine pH,Turbidity,conductivity and temperature. The data preprocessing access the dataset. With the use of this we predict the quality water.
NFR-5	**Availability**	By developing and deploying resilient hardware and software we can analyze the drinking water .
NFR-6	**Scalability**	This project used to measure and determine the quality of water. This provide pollution free and purified water.

CHAPTER-5

PROJECT DESIGN

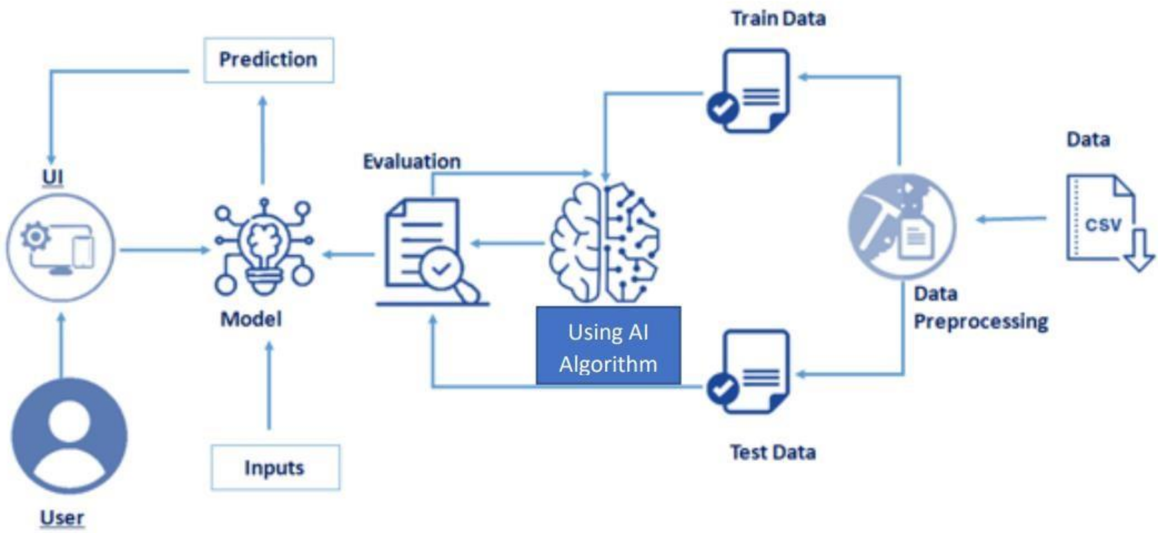
5.1 Data Flow Diagram

Most of the research either employed manual lab analysis, not estimating the water quality index standard, or used too many parameters to be efficient enough. The proposed methodology improves on these notions and the methodology. This helps to achieve the project purpose by step by step procedure.



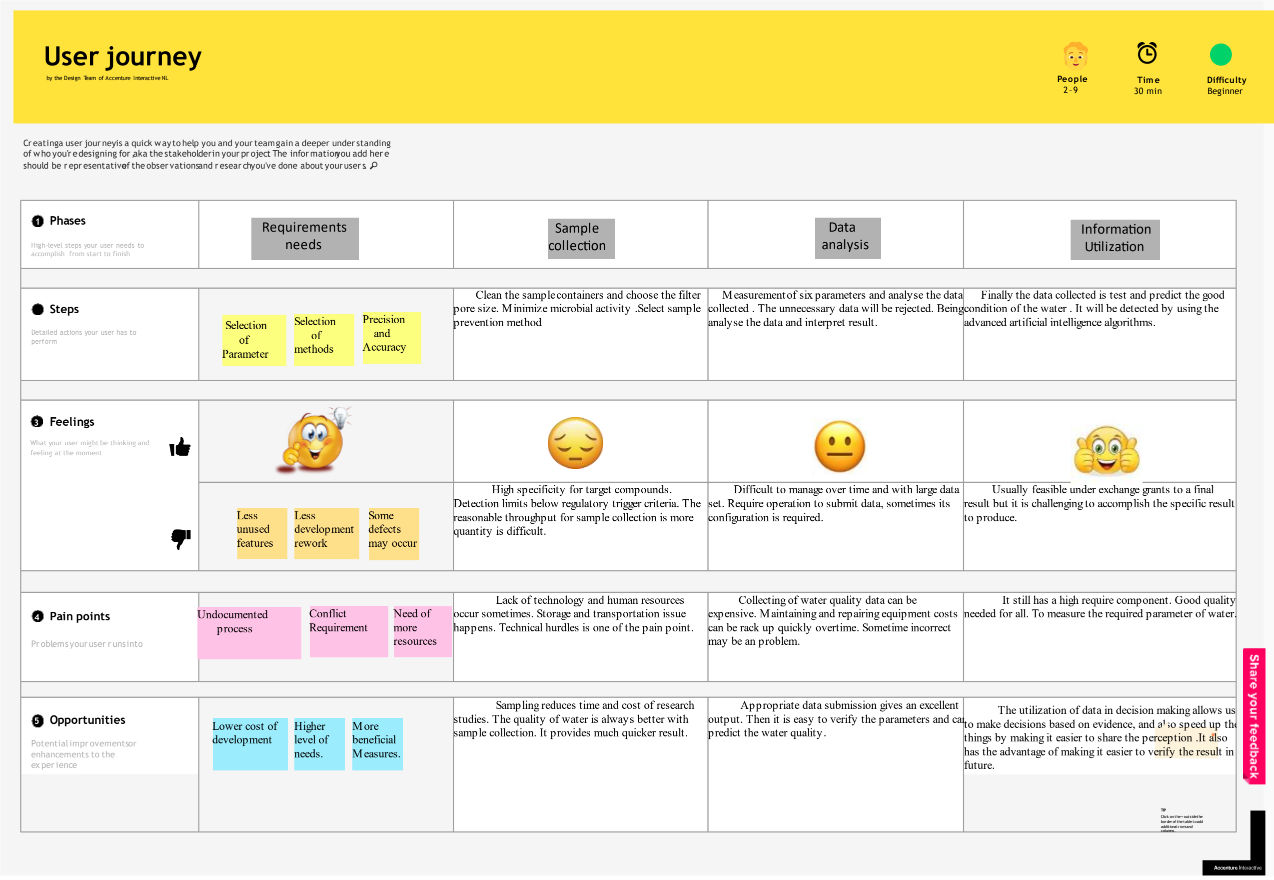
5.2 Solution & Technical Architecture

Technical Architecture:



COMPONENTS	DESCRIPTION	TECHNOLOGIES
User Interface	Userinteract with application	HTML,CSS etc..
Application logic	Logic for a process in application	Java/Python,IBM Watson
Database	Dataset, configuration etc.	MYSQL,NoSQL etc..
Cloud Database	Database service on cloud	IBM DB2,IBM Cloudant
File storage	File storage requirements	IBM Block storage or other storage services
Machine learning model	Purpose of ML model	Recognition model

5.3 User Stories



CHAPTER-6

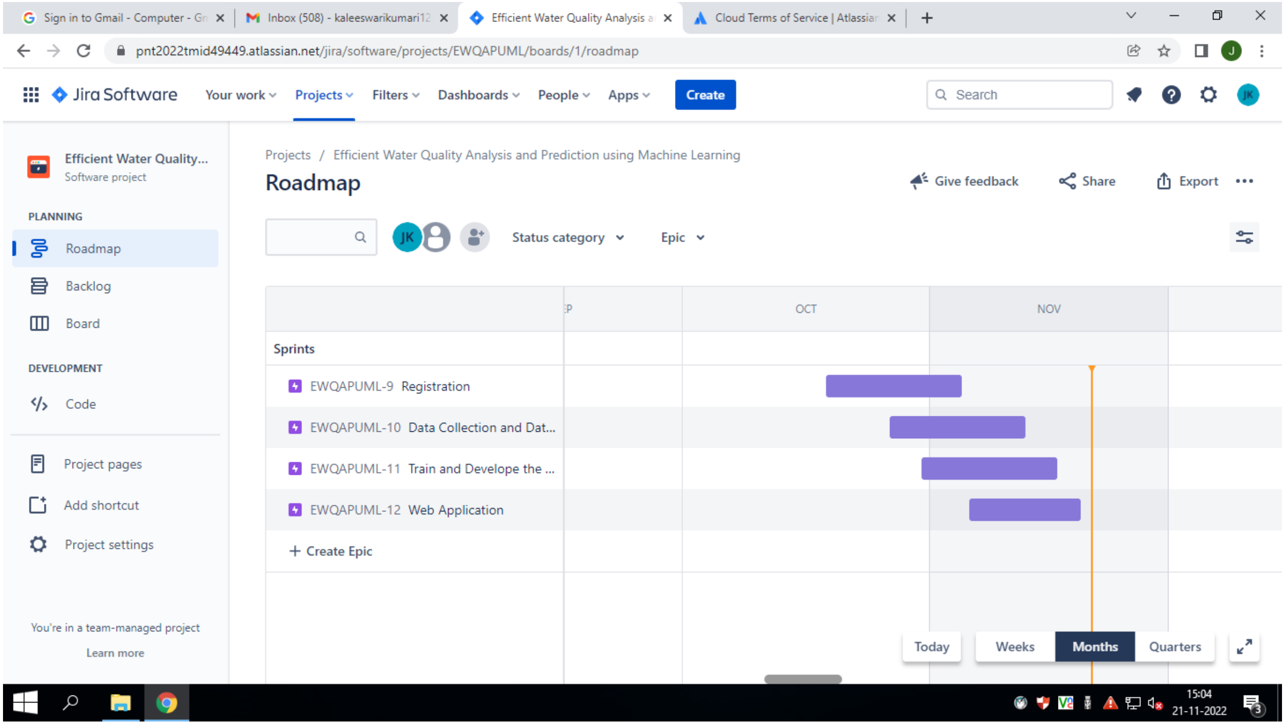
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	K Harinee K KrishnaPriya
Sprint-1	User Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	Medium	S P Subani
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	2	High	S Rishwana Mumtaj M JeyaKaleeswari
Sprint-2	Interface Sensor	USN-1	A sensor interface is a bridge between a device and any attached sensor. The interface takes data collected by the sensor and outputs it to the attached device.	2	High	Jeyakaleeswari S P Subani
Sprint-3	Coding (Accessing datasets)	USN-1	Coding is a set of instructions used to manipulate information so that a certain input results in a particular output.	2	High	K Krishnapriya K Harinee S Rishwana Mumtaj M Jeyakaleeswari
Sprint-4	Web Application	USN-1	As as user, I will show the current Information of the water quality.	1	Medium	S P Subani K KrishnaPriya

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	24 Oct 2022	27 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	28 Oct 2022	01 Nov 2022	20	04 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	09 Nov 2022	20	11 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	18 Nov 2022	20	19 Nov 2022

6.3 Reports From JIRA



CHAPTER-7

CODING AND SOLUTION

7.1 FEATURES:

INDEX:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- The above 3 meta tags *must* come first in the head; any other head content
must
come *after* these tags -->
 <title>Water Quality Prediction</title>
 <link href="https://fonts.googleapis.com/css?family=Lato:400,700"
rel="stylesheet">
 <link type="text/css" rel="stylesheet" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}" />
 <link type="text/css" rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}"
/>
 <style>
 #booking
 {
font-family: 'Lato', sans-serif;
 background: url(../static/background.jpg);
background-size: cover;
background-position: center;
color: #2c3b40;
 }
 </style>
 <!--<img src="{{url_for('static', filename='background.png')}}" />-->
 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries
-->
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
 <!--[if lt IE 9]>
 <script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
 <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
 <!-- ==== GSAP CDN ==== -->
 <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.8.0/gsap.min.js"></script>
 <!-- ==== SCRIPT.JS ==== -->
 <script src="{{url_for('static', filename='script.js' )}}" defer></script>
</head>
```

```
<body style="background-color:#00ccff">
<div id="booking" class="section">
 <div class="section-center">
 <div class="container">
 <div class="row">
 <div class="col-md-4">
 <div class="booking-cta">
 <h1>Water Quality Prediction</h1>
 <p>Let's Find the potability of the water</p>
 </div>
 </div>
 <div class="col-md-7 col-md-offset-1">
 <div class="booking-form">
 <form action={{ url_for("predict") }} method="post">
 <div class="row">
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">ph:</span>
 <input type="text" class="form-control" name="ph"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Year>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Hardness: </span>
 <input type="text" class="form-control" name="Hardness"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Month>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Solids:</span>
 <input type="text" class="form-control" name="Solids"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <! Date>
 </div>
 <div class="row">
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Chloramines:</span>
 <input type="text" class="form-control" name="Chloramines"
placeholder="Enter value" required="true">
```

```
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Year>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Sulfate:</span>
 <input type="text" class="form-control" name="Sulfate"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Month>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Conductivity:</span>
 <input type="text" class="form-control" name="Conductivity"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <! Date>
 </div>
 <div class="row">
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Organic_carbon:</span>
 <input type="text" class="form-control"
name="Organic_carbon" placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Year>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Trihalomethanes:</span>
 <input type="text" class="form-control"
name="Trihalomethanes" placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Month>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">Turbidity:</span>
 <input type="text" class="form-control" name="Turbidity"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
```

```
 </div>
 <! Date>
 </div>
 <div class="row">
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">nph:</span>
 <input type="text" class="form-control" name="nph"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Year>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">nHardness:</span>
 <input type="text" class="form-control" name="nHardness"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Month>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">wph:</span>
 <input type="text" class="form-control" name="wph"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <! Date>
 </div>
 <div class="row">
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">wHardness:</span>
 <input type="text" class="form-control" name="wHardness"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <!Year>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">wSolids:</span>
 <input type="text" class="form-control" name="wSolids"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
```

```html
 </div>
 <!Month>
 <div class="col-md-4">
 <div class="form-group">
 <span class="form-label">wqi:</span>
 <input type="text" class="form-control" name="wqi"
placeholder="Enter value" required="true">
 <span class="select-arrow"></span>
 </div>
 </div>
 <! Date>
 </div>
 <!--Year,Month,Date end-->
 <div class="form-btn">
 <button class="submit-btn">Predict</button>
 </div>
 <!Button>
 {% if prediction_text == 0 %}
 <h2 >" Not Potable, the water can't be drink "</h2>
 {% elif prediction_text == 1 %}
 <h2 >" Potable, the water can be drink "</h2>
 {% endif %}
 </center>
 </form>
 <!--Form end-->
 </div>
 <!Booking form>
 </div>
 </div>
 </div>
 </div>
 </div>
 </p>
</body>
</html>
```

APP.PY:

```python
from flask import Flask, render_template, request
import requests
import requests
# NOTE: you must manually set API_KEY below using information retrieved from
your
IBM Cloud account.
import requests
# NOTE: you must manually set API_KEY below using information retrieved from
your
IBM Cloud account.
API_KEY = "nFFWACn7pVNTQWlnb7pusoXVa63g0vFEq_8Y2x2pxZSE"
```

```
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)
@app.route('/')
def index():
 return render_template('index.html')
@app.route('/predict', methods = ['POST'])
def predict():

 ph = request.form['ph']
 Hardness = request.form['Hardness']
 Solids = request.form['Solids']
 Chloramines = request.form['Chloramines']
 Sulfate = request.form['Sulfate']
 Conductivity = request.form['Conductivity']
 Organic_carbon = request.form['Organic_carbon']
 Trihalomethanes = request.form['Trihalomethanes']
 Turbidity = request.form['Turbidity']
 nph = request.form['nph']
 nHardness = request.form['nHardness']
 wph = request.form['wph']
 wHardness = request.form['wHardness']
 wSolids = request.form['wSolids']
 wqi = request.form['wqi']



 t =
[[float(ph),float(Hardness),int(Solids),float(Chloramines),float(Sulfate),float(Conduc
tivity),fl
oat(Organic_carbon),float(Trihalomethanes),float(Turbidity),int(nph),int(nHardness),
float(w
ph),float(wHardness),float(wSolids),float(wqi)]]

 payload_scoring = {"input_data": [{"fields":
[["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14"]],
"values":t
}]}

 response_scoring =
requests.post('https://ussouth.ml.cloud.ibm.com/ml/v4/deployments/94d69d4c-4e12-
4662-aba9-
fe193faa3d90/predictions?version=2022-11-07', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})

 print("Scoring response")
```

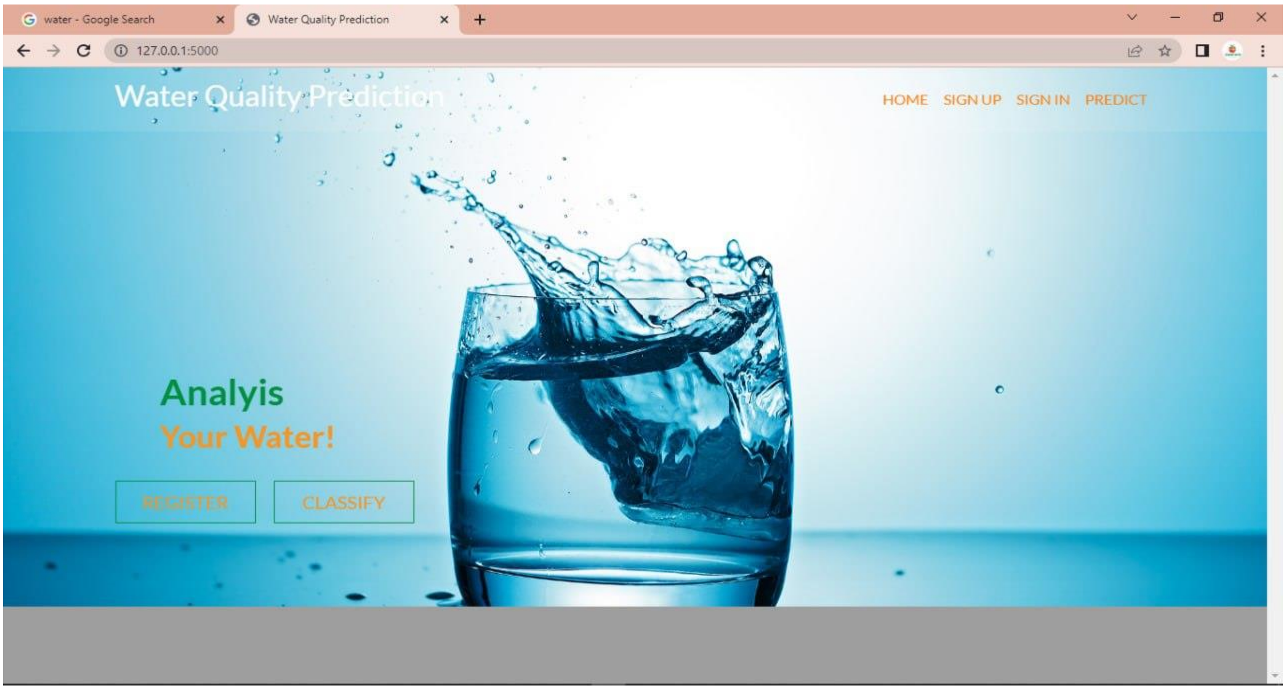
```
 payload_scoring = {"input_data": [{"fields":
[["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14"]],
"values":t
}]}

 pred= response_scoring.json()
 output=pred['predictions'][0]['values'][0][0]
 print(output)

 return render_template("index.html", prediction_text = output)

if __name__=='__main__':
 app.run(debug = False)
```

CHAPTER-8

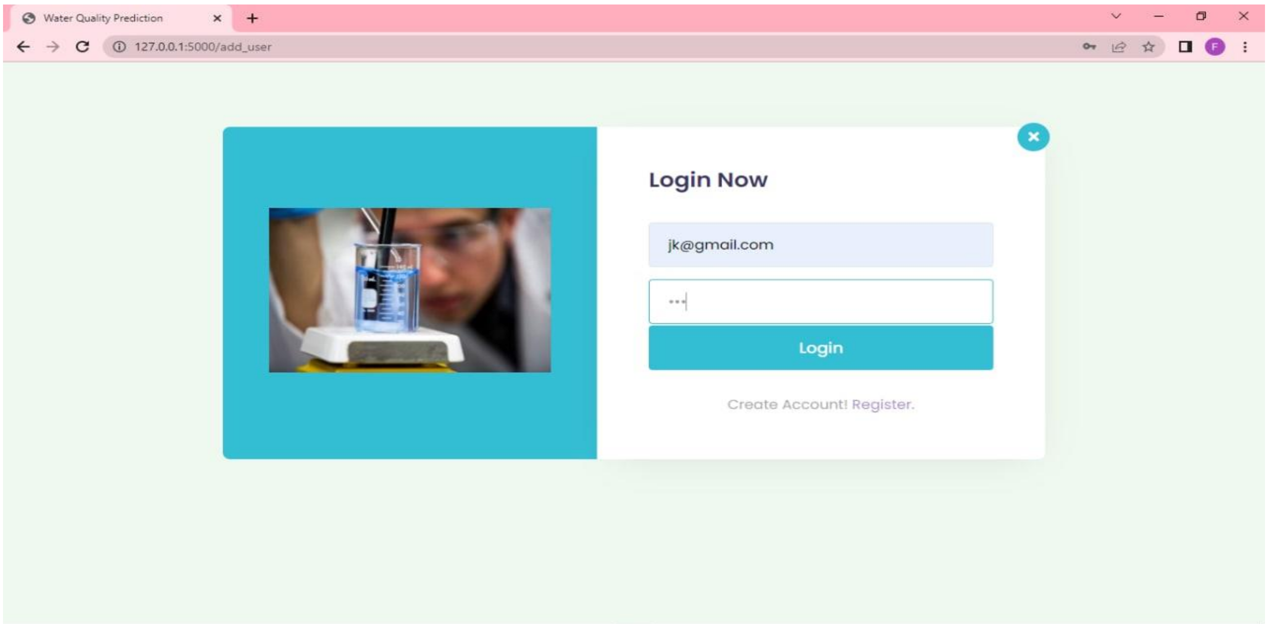
TESTING

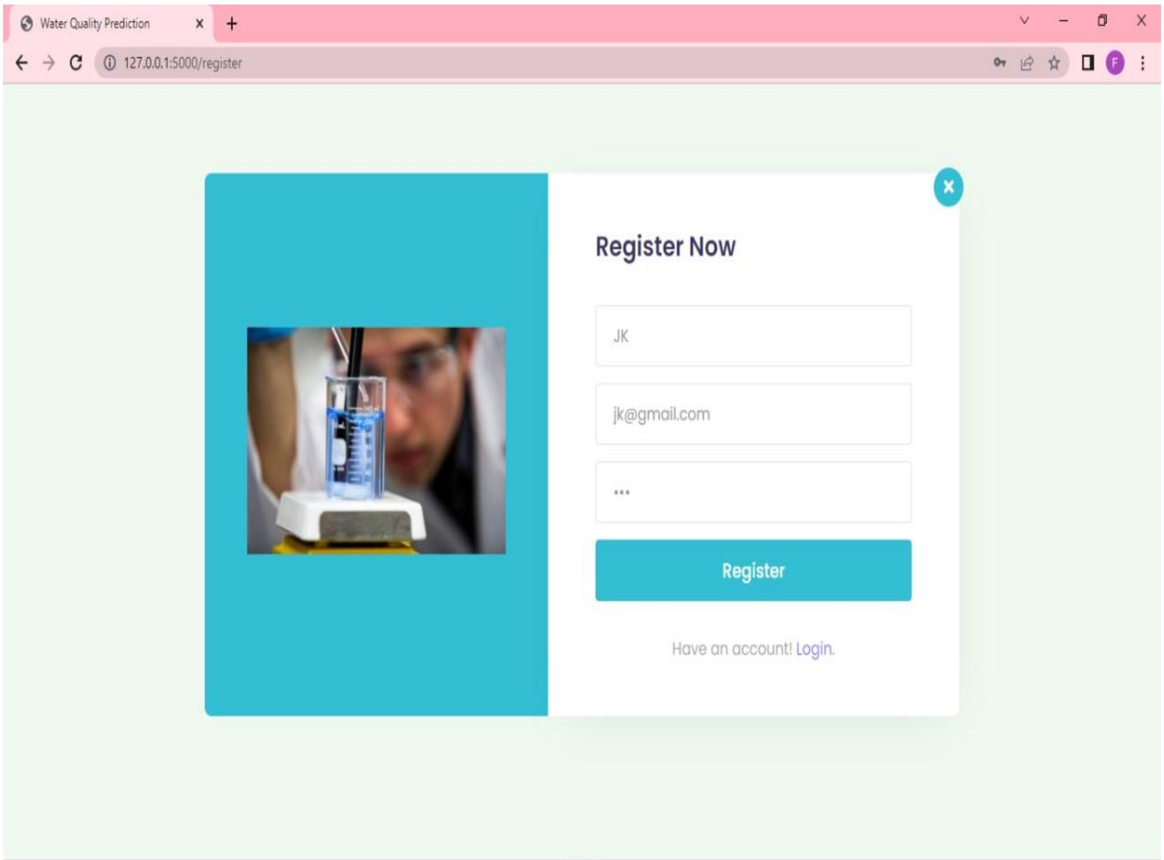
8.1 Test Cases

Home:

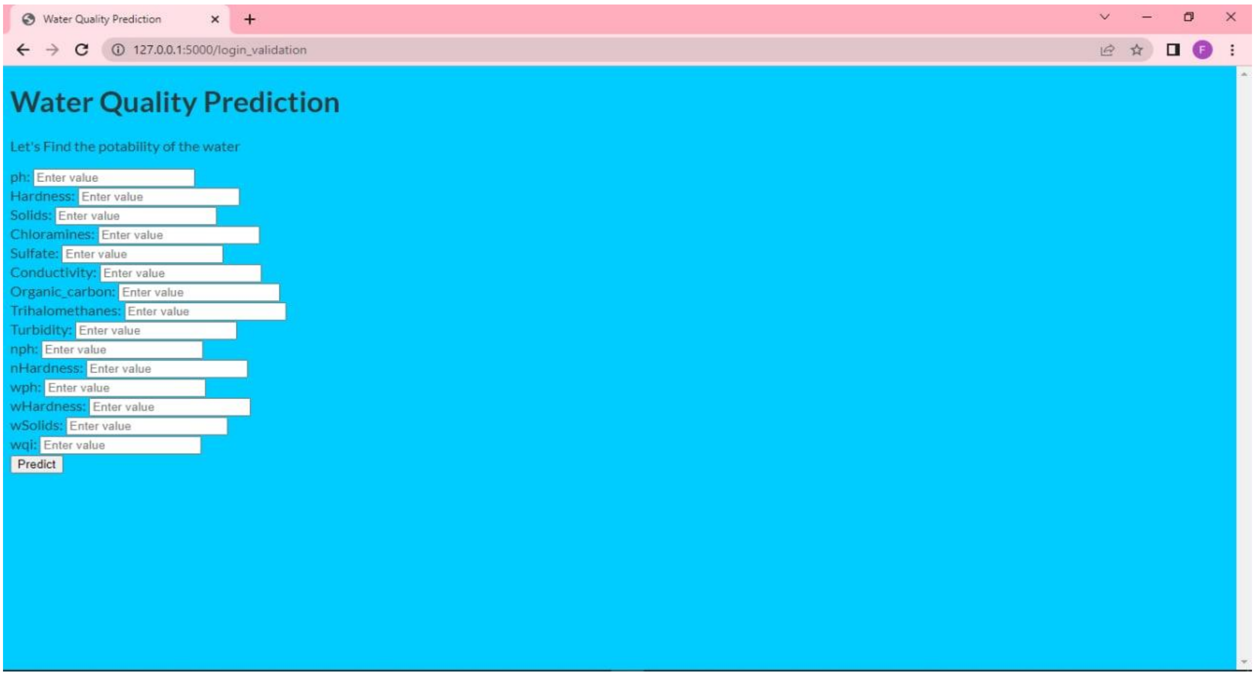


Login:

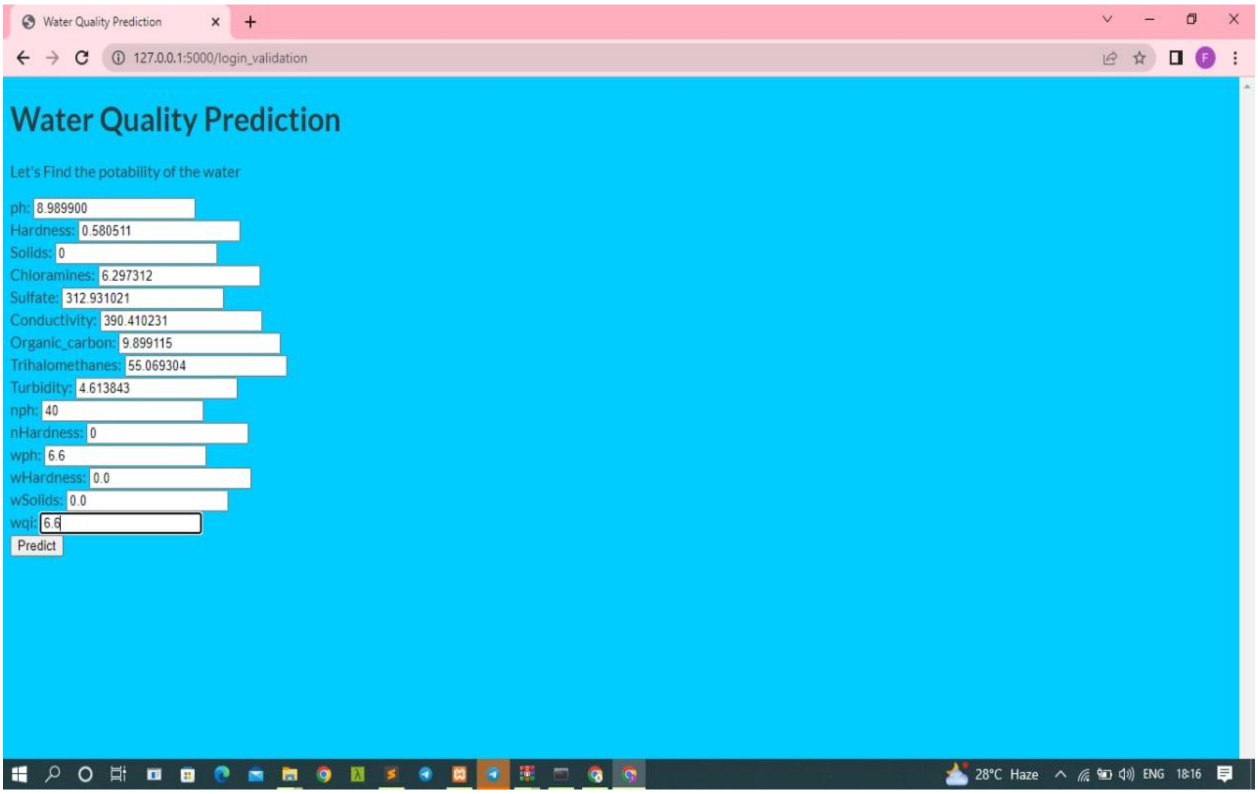




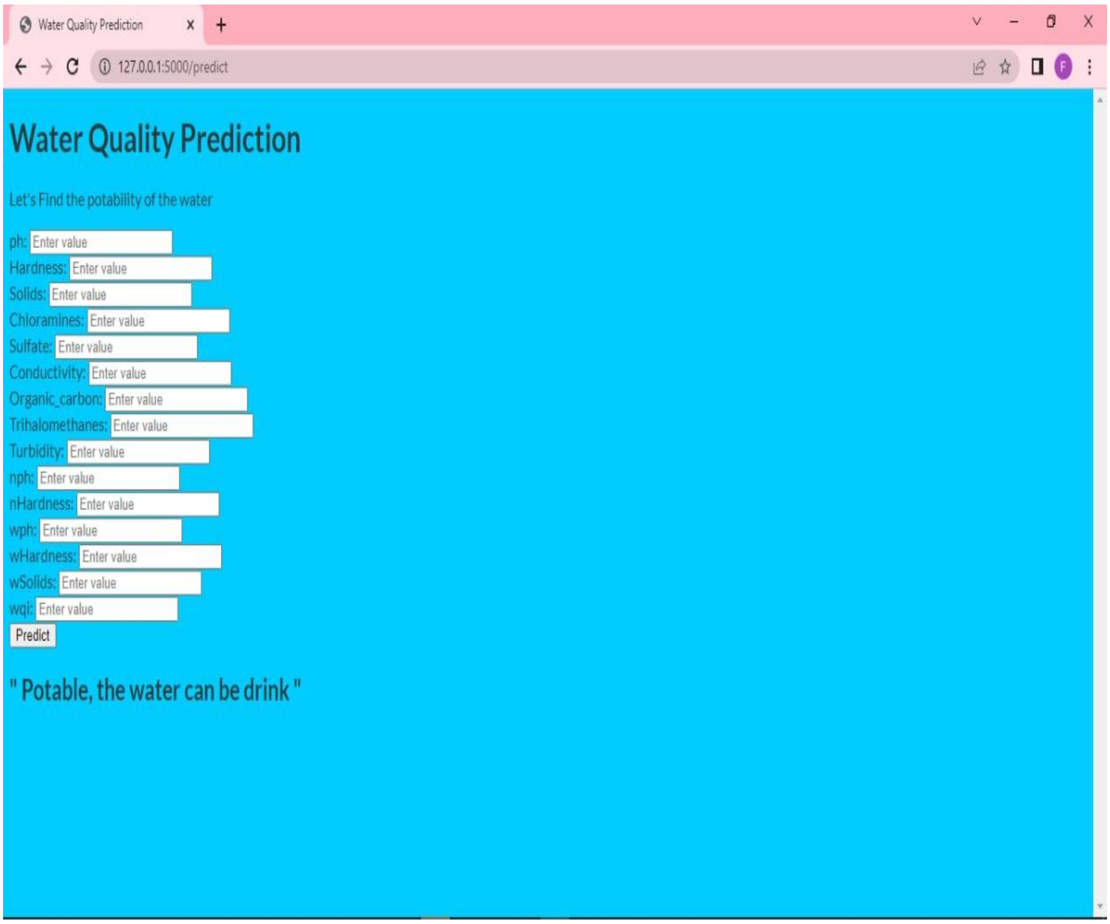
Water quality checking:



Giving values:



Result:



8.2 User Acceptance Testing

1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizer Recommendation System for Disease Prediction] project at the time of the release to User Acceptance Testing (UAT).

2.Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how

they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	3	6	5	21
Duplicate	4	0	3	0	7
External	1	2	0	1	4
Fixed	14	1	3	8	26
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	2	0	6
Totals	26	11	18	19	67

Testcase Analysis:

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	0	7
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9

RESULTS

9.1 Performance metrics

Model Summary

```
0.575 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'best'}
0.573 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'random'}
0.575 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'best'}
0.571 (0.032) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'random'}
0.577 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'best'}
0.578 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'random'}
0.574 (0.033) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'best'}
0.578 (0.023) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'random'}
0.580 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'best'}
0.582 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'random'}
0.576 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'best'}
0.584 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'random'}
0.576 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'best'}
0.585 (0.036) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'random'}
Training Score: 90.11450381679388
Testing Score: 59.29878048780488
```

Accuracy

```
In [28]: prediction=dt.predict(X_test)
         print(f"Accuracy Score = {accuracy_score(Y_test,prediction)*100}")
         print(f"Confusion Matrix =\n {confusion_matrix(Y_test,prediction)}")
         print(f"Classification Report =\n {classification_report(Y_test,predict

         Accuracy Score = 56.859756097560975
         Confusion Matrix =
          [[274 128]
          [155  99]]
         Classification Report =
                       precision    recall  f1-score   support

                    0       0.64      0.68      0.66       402
                    1       0.44      0.39      0.41       254

             accuracy                           0.57       656
            macro avg       0.54      0.54      0.54       656
```

CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1Advantage:

* Easily detect the water quality.

* Most Accurate

* Flexible Model which can give maximized outcome

* No Specific Requirements needed to implement the model

10.2 Disavantage:

*****If the given data not related to the dataset then it gives an fault result**.**

*****The data of water quality is modified accordingly.

*More data needed for the quality checking.

CHAPTER-11

CONCLUSION

The performance of machine learning technique to predict the water quality components of an Indian water quality dataset was evaluated in this work. The most well-known dataset variables, such as Ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, Turbidity were obtained for this purpose. The findings revealed that the applied models performed well in forecasting water quality parameters; however, the greatest performance was linked with the MLR with Accuracy Upper. Further research will be done to build models that combine the proposed method with other techniques and deep learning approaches to improve the efficacy of the selection process.

CHAPTER-12

FUTURE SCOPE

In future works, we propose integrating the findings of this research in a large-scale IoT-based online monitoring system using only the sensors of the required parameters. The tested algorithms would predict the water quality immediately based on the real-time data fed from the IoT system. The proposed IoT system would employ the parameter sensors of pH, turbidity, temperature and TDS for parameter readings and communicate those readings using an Arduino microcontroller and ZigBee transceiver. It would identify poor quality water before it is released for consumption and alert concerned authorities. It will hopefully result in curtailment of people consuming poor quality water and consequently de-escalate harrowing diseases like typhoid and diarrhea. In this regard, the application of a prescriptive analysis from the expected values would lead to future facilities to support decision and policy makers.

CHAPTER-13

APPENDIX

Github: https://rb.gy/fweu1o

Demo Link: https://bit.ly/3gmHKSD