Project Report

Efficient Water Quality Analysis and Prediction using Machine Learning

By,

G. Sree Harine (Team Leader)

K. Rohith Gupta (Team Member)

K. Ramnath (Team Member)

L. Santhosh Srinivas (Team Member)

INDEX

1. INTRODUCTION

1.1 Project Overview:

The consistent increase in the rate of growth of India's population has also led to the increase in demand for water, particularly in the urban and suburban areas where the rate of increase is higher compared to rural areas. The problem of access to safe drinking water and sanitation facilities in urban areas of India is also a major concern. At present, 163 million people do not have access to safe drinking-water and 210 million people lack access to improved basic sanitation in India. There is a need for water quality testing - many physical and chemical tests are available and conducted in industries to test the quality of water. Available data suggests that pollution levels have increased in surface water as well as groundwater. More than 100 million people in suburban areas are exposed to poor water quality. The lack of sufficient infrastructure, services and funds to support water and wastewater treatment facilities required for an urban area further exacerbates the problem.

1.2 Purpose

Even though there are physical and chemical tests available to measure the water quality, there is no guarantee that all the locals can use those facilities to test the quality of water they are utilizing for domestic use. Modern chemistry, we can detect thousands of chemicals in water, even at extremely low concentrations. Machine learning techniques can be helpful in efficiently predicting the quality of water in different locations; they can also be conveniently used by people without the use of any expensive equipment. The water quality can be predicted and analyzed anywhere anytime in the absence of sophisticated instruments.

Therefore, machine learning aids in predicting the quality remotely by common people and can be utilized to find out the condition of water in their location.

2. LITERATURE SURVEY

2.1 Existing methodologies and References

Sl no.	Literature Paper	Author	Inference
1	Assessment of drinking water quality:A case study of Ambala cantonment area,Haryana , India	Routh Chadetrik,Sharma Arabindha	Water is a vital resource for human survival. In the present study, the physicochemical characteristics of groundwater of Ambala Cantonment area were assessed for its suitability for drinking purposes. A total of 26 water samples were collected from deep aquifer based tube wells from different parts of Ambala Cantonment area. In order to assess the ground water quality, the water samples were analyzed for different physicochemical properties, e.g., pH, electrical conductivity (EC), total dissolved solids (TDS), calcium, magnesium, total harness (TH), sodium, potassium, carbonate, bicarbonate, total alkalinity (TA), chloride, fluoride and sulphate concentrations. The results were compared with the standards prescribed by World Health Organization (WHO) and Bureau of Indian Standard (BIS). All the physiochemical parameters were found to be in the prescribed permissible limit. The correlation matrix was also calculated for different parameters of drinking water. From the pH values it is clear that the ground water of the study area is alkaline in nature and the total hardness varies in between 116.6129.4 mg/l, which indicates that water in the deep aquifer is moderately hard. Hence it is suggested to the cantonment localities to soften the tube well water before consumption.
2	Impact of Irrigation Water Quality on human health:A case study in India	Jeena.T.Srinivasan,V.Ratna Reddy	It is seen that higher rates of morbidity exist in the wastewater irrigated villages when compared to the control village. Specifically, adult and female morbidity rates are significantly higher than child and male morbidity rates. From the logit analysis it is seen that exposure to wastewater and engagement in activities based on it places the households in higher risk groups to report morbidity. Small and marginal farmers incur higher economic cost of illness. However, it has

			been difficult to ascertain the cause and effect relationship as most of the households have reported more common illnesses like fever, head ache, skin itching, stomach ailments, etc. The study points out the need for a comprehensive risk assessment and adoption of risk management measures including setting standards for treatment and discharge of wastewater and regulations on the type of uses etc. to prevent unplanned use of untreated or partially treated wastewater while taking into account local conditions.
3	Impact of informal regulation of pollution on water quality in rivers in India	Nandini Banarjee, Bishwanath Goldar	In this paper an attempt is made to assess the impact of informal regulation of water pollution on water quality in Indian rivers. For this purpose, an econometric analysis of determinants of water quality in Indian rivers is carried out using water quality (water class) data for 106 monitoring points on 10 important rivers for five years, 1995–1999. To explain variations in water quality, an Ordered Probit model is estimated, in which poll percentage in parliamentary elections, a proxy for the intensity of informal regulation, is taken as one of the main explanatory variables. Rainfall, industrialization, irrigation intensity and fertilizer use are some of the other explanatory variables used in the model to control for the influence of these factors. As expected, river water quality is found to be positively related with rainfall, and negatively related with industrialization, irrigation intensity and fertilizer use. A significant positive relationship is found between poll percentage and water quality, and also between the rate of increase in literacy level in a district and the water quality in rivers flowing through the district. These results point to a significant favorable effect of informal regulation of pollution on water quality in rivers in India.
4	Challenges of sustainable water quality management in rural India	R.Srikanth	High rates of mortality and morbidity due to water-borne diseases are well known in India. Serious degradation of water quality in urban India has often been attributed to indiscriminate disposal of sewage and

			industrial effluents into surface water bodies. Although some degree of intervention in terms of chlorination and monitoring of water quality exists in major cities and towns, rural India, which constitutes the bulk (70%) of the population, is usually deprived of such interventions. The population in rural India is mainly dependent on the groundwater as a source of drinking water. As a quality concern the groundwater is often found to be contaminated with fluoride, arsenic, iron and salts. In recent years, fluorosis has emerged as major public health issue in rural India. At the technical level, some progress has been made in the development and use of field-level diagnostic kits. Decentralization of health-related monitoring at the villages needs to be institutionalized and this requires capacity development at all levels. This article discusses the various components that impact effective water quality management in rural India. Experience suggests that redesigning of data management programme at village, district and at national level, upgradation of district-level laboratories and addressing technical, legal and institutional components should become the first steps in achieving effective water-quality management and providing better health to millions of people living in rural India.
5	A study on water quality prediction by a hybrid CNN-LSTM model with attention mechanism	Yurong Yang, Qingyu Xiong, Chao Wu, Qinghong Zou, Yang Yu, Hualing Yi & Min Gao	The water environment plays an essential role in the mangrove wetland ecosystem. Predicting water quality will help us better protect water resources from pollution, allowing the mangrove ecosystem to perform its normal ecological role. New approaches to solve such nonlinear problems need further research since the complexity of water quality data and they are easily affected by the noise. In this paper, we propose a water quality prediction model named CNN-LSTM with Attention (CLA) to predict the water

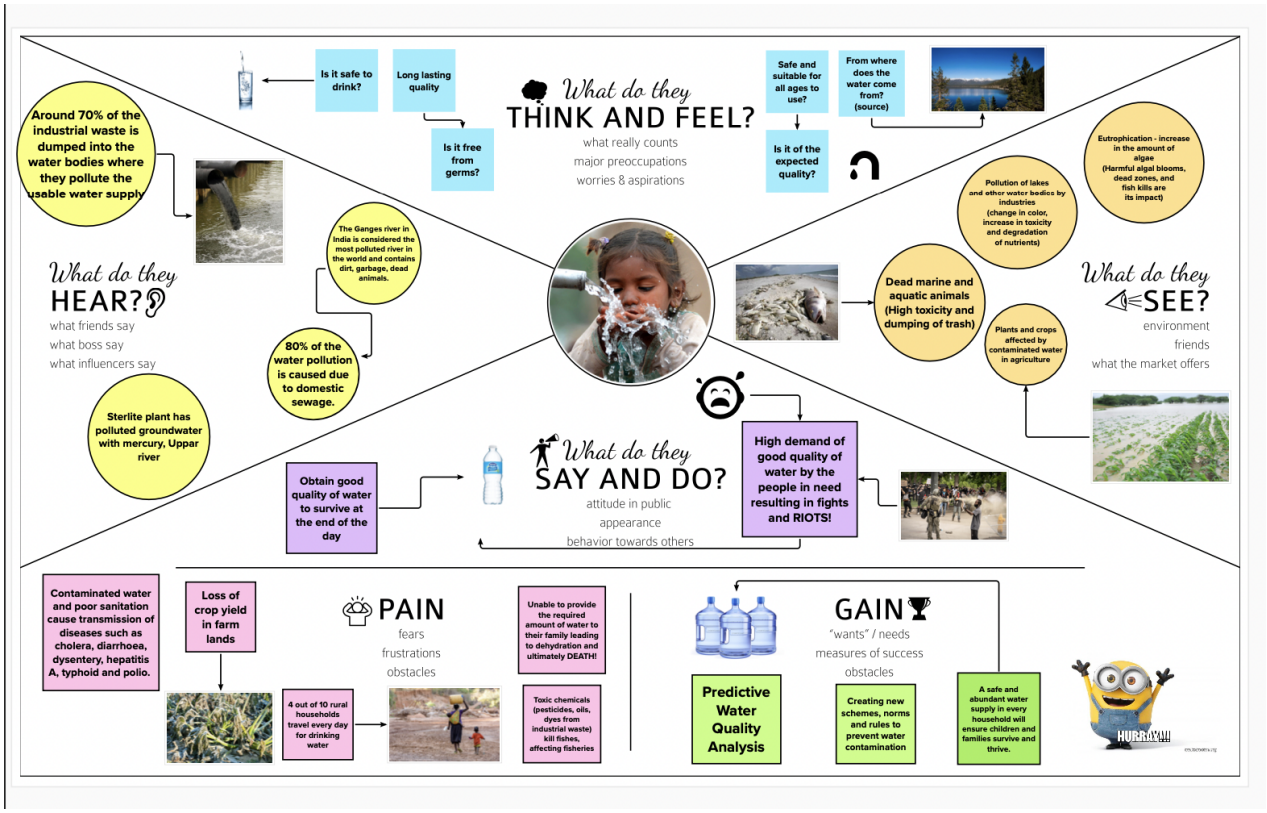
			quality variables. We conduct a case study on the water quality dataset of Beilun Estuary to predict pH and NH_3-N. Linear interpolation and wavelet techniques are used for missing data filling and data denoising, respectively. The hybrid model CNN-LSTM is highly capable of resolving nonlinear time series prediction problems, and the attention mechanism captures longer time dependence. The experimental results show that our model outperforms other ones, and can predict with different time lags in a stable manner.

2.2 Problem Statement Definition

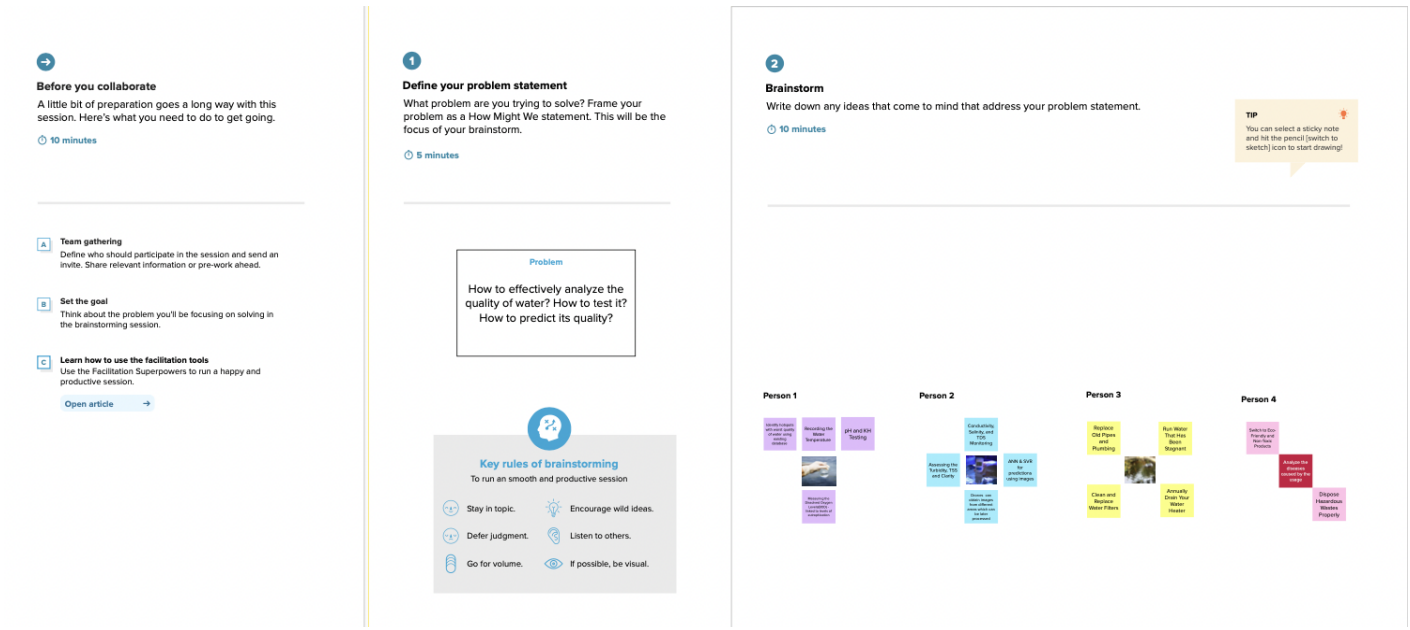
To predict the quality of water using machine learning techniques by calculating the water quality index focusing on the suburban areas where people face a lot of water quality issues - eg: dumping of domestic sewage, effluents from chemical industries etc.

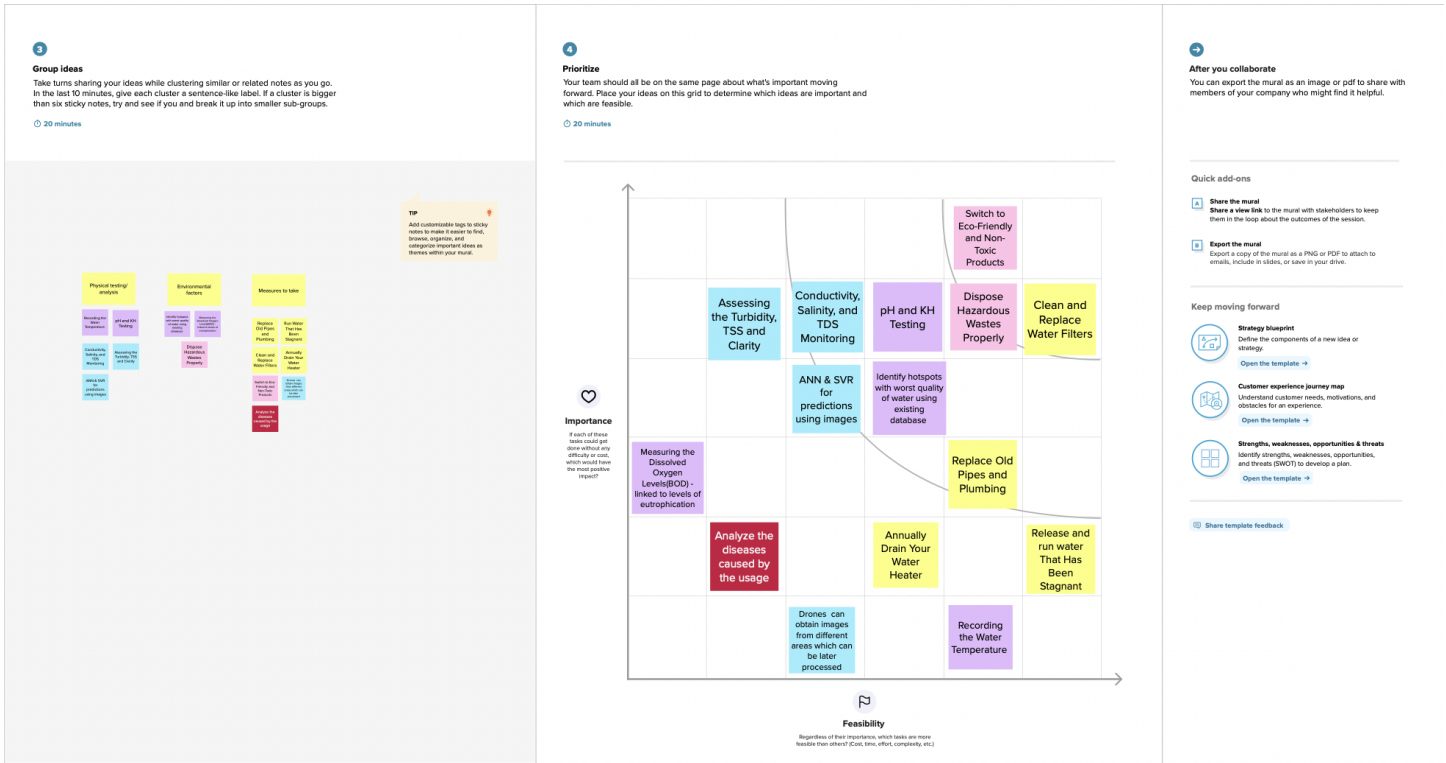
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

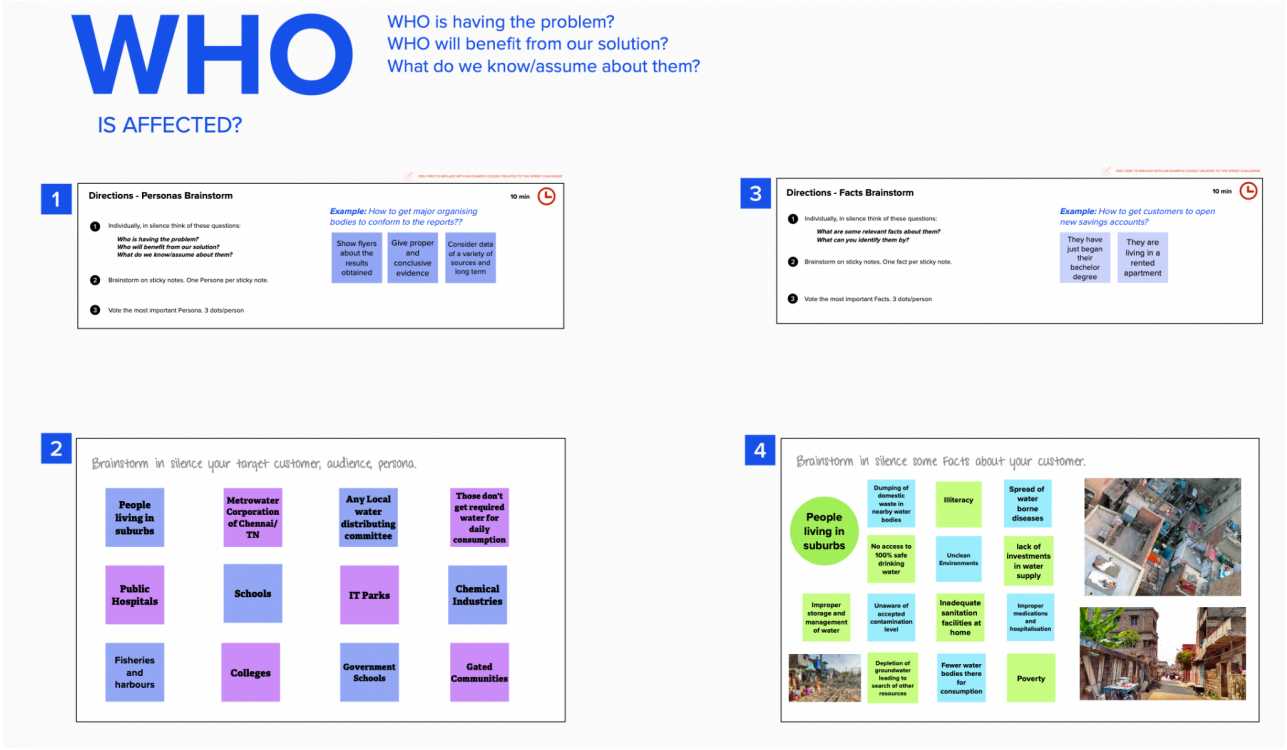




3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To analyse and predict the water quality efficiently using Machine Learning techniques.
2.	Idea / Solution description	A web application will be created which will predict the quality of water using the water quality index that is calculated after processing the dataset. The data will be trained and tested using python and ML techniques.
3.	Novelty / Uniqueness	Using ML techniques to predict the quality instead of using physical measurements to obtained the quality of water in different places. ML techniques improve the accuracy of measurement over chemical techniques which may not be feasible to obtain all the required features to predict the water quality. Physical and chemical measurements may lead to the usage of expensive instruments. ML techniques make the process easier and feasible.
4.	Social Impact / Customer Satisfaction	Our target audiences are people in suburban areas. Through efficient prediction of water quality is areas where water is not abundant, it helps in identifying water pollution, provides safe drinking water to all, more sanitation facilities will be available and customers tend to manage and use water in an efficient manner that will increase the awareness of water borne diseases and thus reduce their spread. When quality is predicted in an efficient manner, clean water will be available for both domestic and agricultural usage by people.
5.	Business Model (Revenue Model)	More industries that provide sanitation facilities and products (like water purifiers, quality testers etc.) will be put up, more waste water treatment plants, better insights in health concerns and there may also be an increase in awareness and demand for better water quality testing and availability. People will start looking for treatments related to water-borne diseases as the awareness increases.
6.	Scalability of the Solution	The proposed solution is feasible and scalable which can be used both by industries (for testing their raw materials for their process) and even by common people – not as expensive as the physical instruments used to measure the quality of water.

3.4 Problem statement ideation



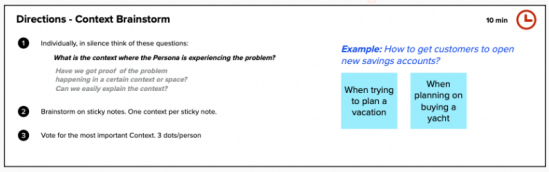
WHERE/WHEN

DOES IT OCCUR?

What is the context where the Persona is experiencing the problem?

Can we easily explain the context?

Have we got proof of the problem happening in a certain context or space?

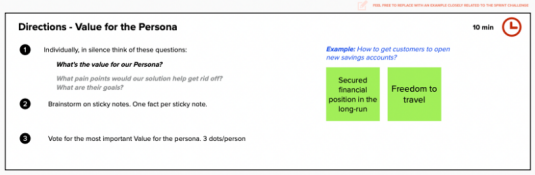


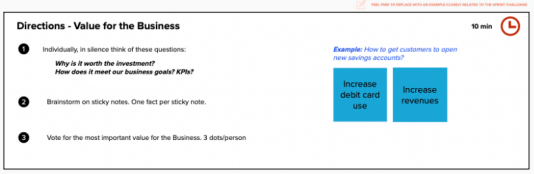
WHY

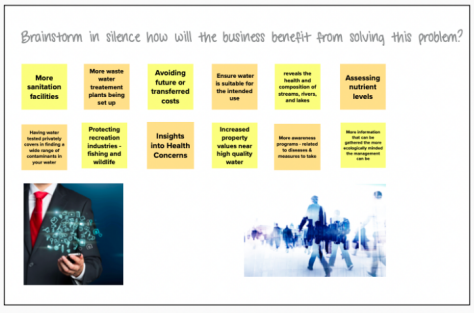
CARE ABOUT IT?

What is the most important value for the user?
What pain points would a solution help get rid of?

Why is it worth our investment?
How does it meet our business goals? KPIs?







Problem Statement

ONCE THE TEAM HAS DEFINED THE PROBLEM, TRANSFER THEIR OUTPUT IN THE TEXT BOXES BELOW, THEY SERVE AS THE SKELETON OF THE PROBLEM STATEMENT.

WHO?

People living in suburbs

More people in the suburbs are affected due to poor water quality.

WHAT?

Industries dumping waste into the water resources

Industries tend to dump dyes, chemicals and other effluents (wastes) from their industrial processes into the water bodies.

WHERE/ WHEN?

Near dyes and chemical industries

More concentration of chemicals released directly into water bodies near the suburban areas.

WHY?

Safe drinking water for all - urban and suburban

Customer value/benefit

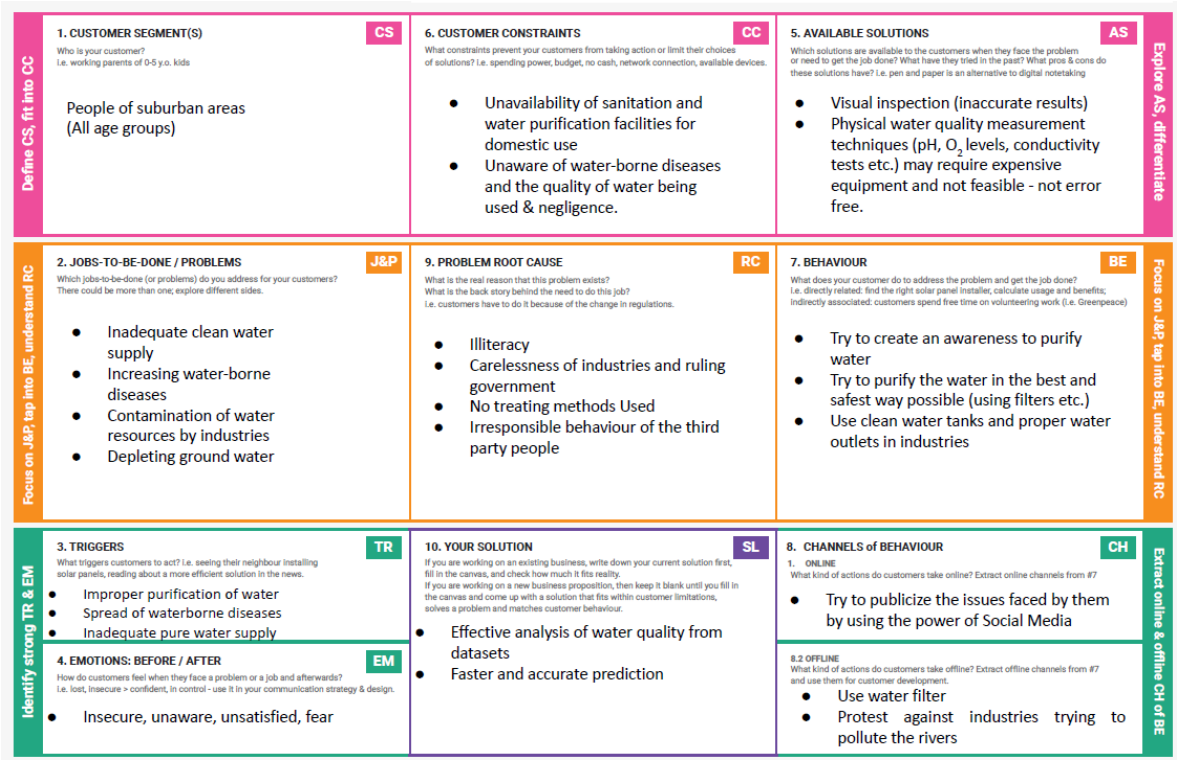
Safe drinking water leads to reduction in water borne diseases and for a healthier lifestyle.

More sanitation facilities

Business value/benefit

Setting up of new industries in the field of sanitary products and better water treatments.

3.4 Problem solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

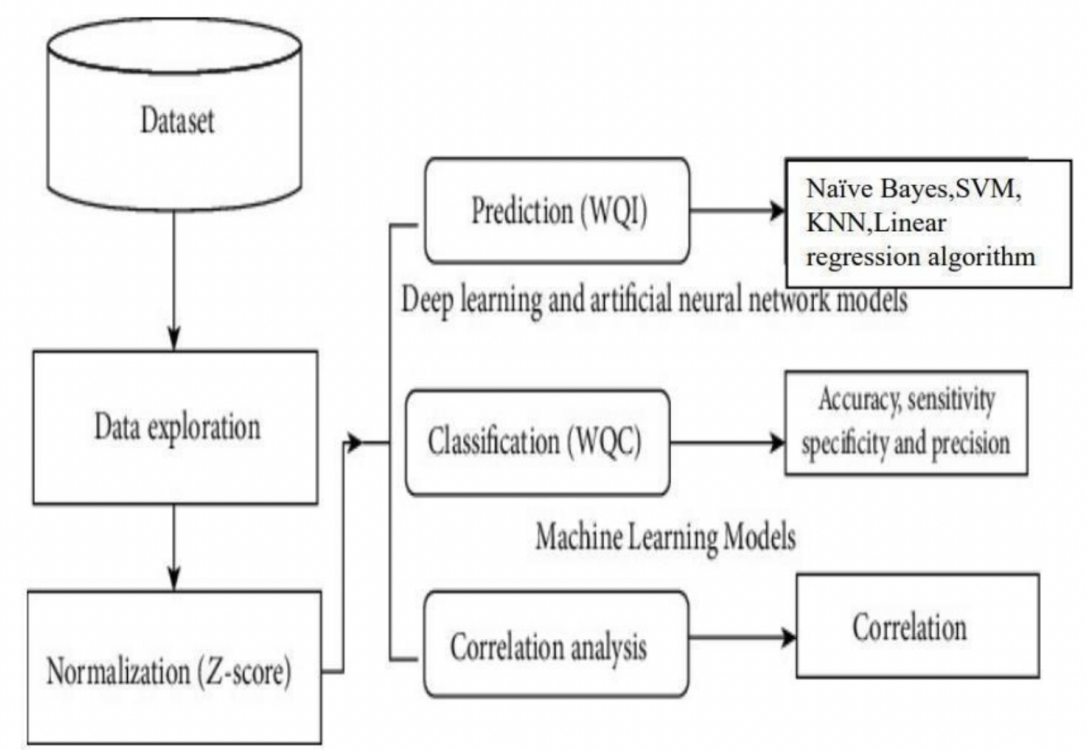
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input (parameters for testing)	7 input parameters fed as input (BOD, DO, pH, Conductivity, Year, Total Coliform, NITRITENANN (mg/l).
FR-3	Reports and results	1.Result of the water quality analysis will be sent a message to the user. 2.The real-time water quality report is collected and the dataset is used to predict the water quality for future works.
FR-4	Prediction	Water Quality Index (WQI) formula will be used for the water quality analysis and prediction. WQI value is displayed on the application page.

4.2 Non-Functional requirements

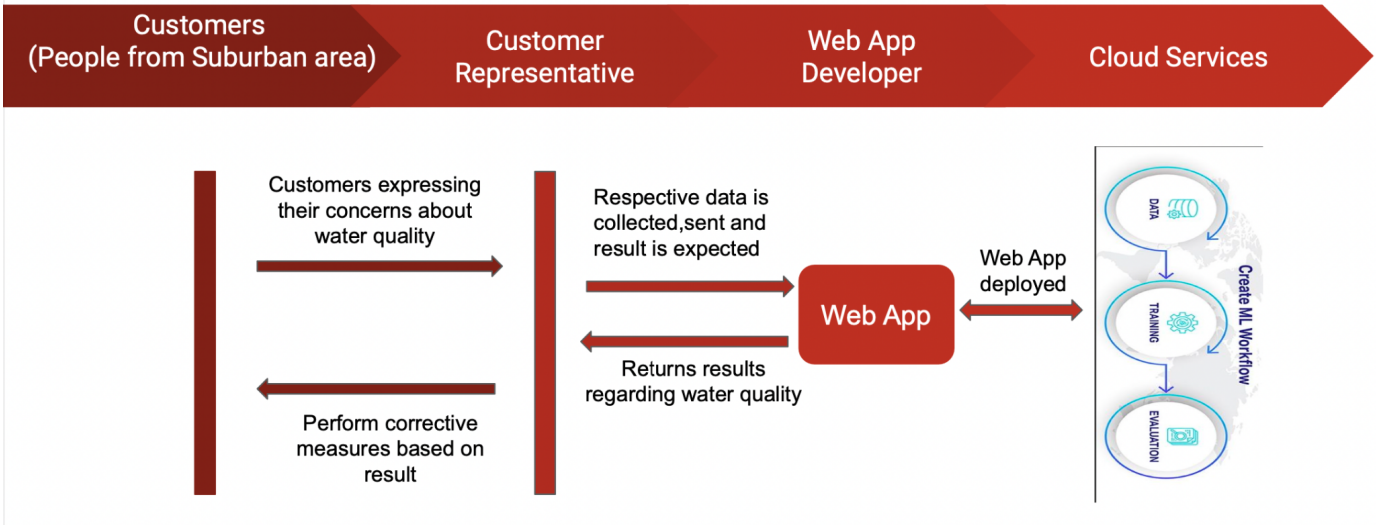
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Allows users to identify missing data elements available in the water quality portal data.
NFR-2	**Reliability**	Consistent results will be obtained even after multiple use (results and prediction based on the dataset).
NFR-3	**Performance**	The system effectively compares the input parameters given by the users with the dataset.
NFR-4	**Availability**	Even if there is infrastructure failure, the model will continue to function.
NFR-5	**Scalability**	Water Quality Index (WQI) and Water Quality Classification (WQC) are accurately predicted for different geographic locations with the available data.

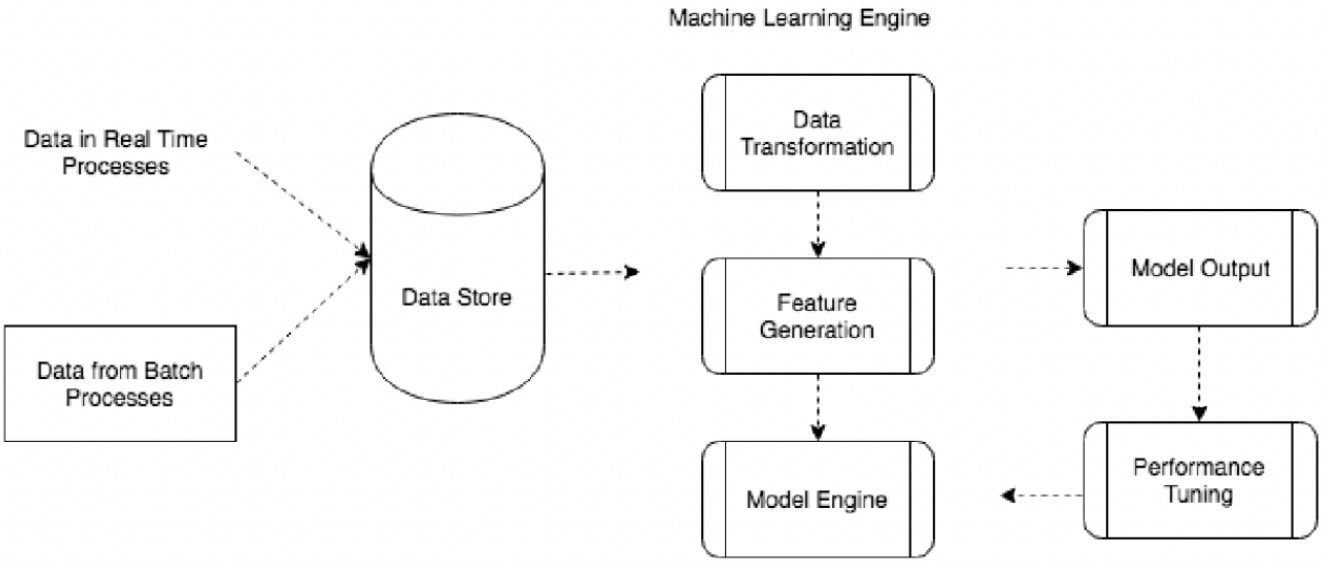
5. PROJECT DESIGN

5.1 Data Flow Diagrams

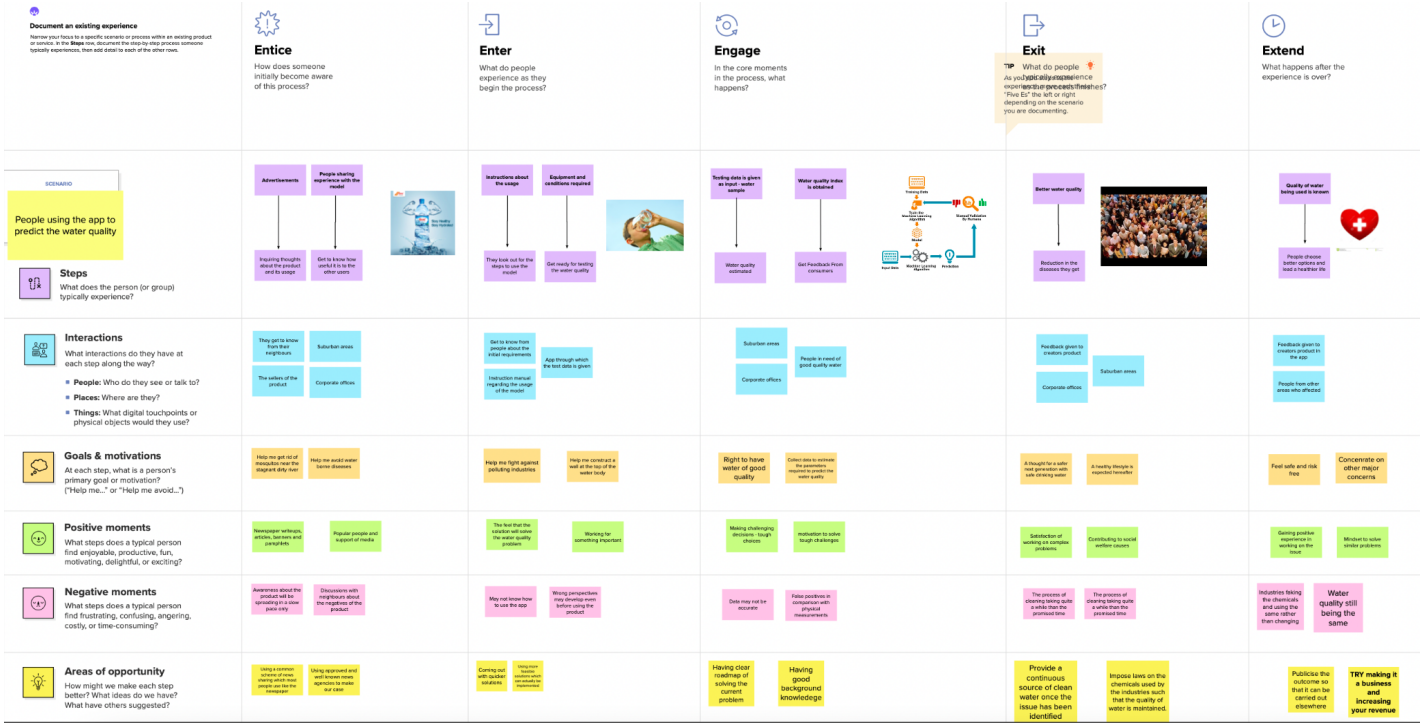


5.2 Solution & Technical Architecture





5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

S.NO	MILESTONE	DESCRIPTION	DURATION	Working Status
1	Prerequisites	Prerequisites are all the needs at the requirement level needed for the execution of the different phases of a project.	1 WEEK	Completed
2	Ideation	To generate ideas and solutions through sessions such as Sketching, voting the best of alternatives, Prototyping, Brainstorming and comparison with other techniques	1 WEEK	Completed
3	Project design phase	Project design is an early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out. The aim is to develop one or more designs that can be used to achieved the desire goals.	1 WEEK	Completed
4	Project Planning Phase	In this phase we create the technical design,functional requirements,tech stack, task list, resources ,communication plan,budget and initial schedule for project.	1 WEEK	Completed

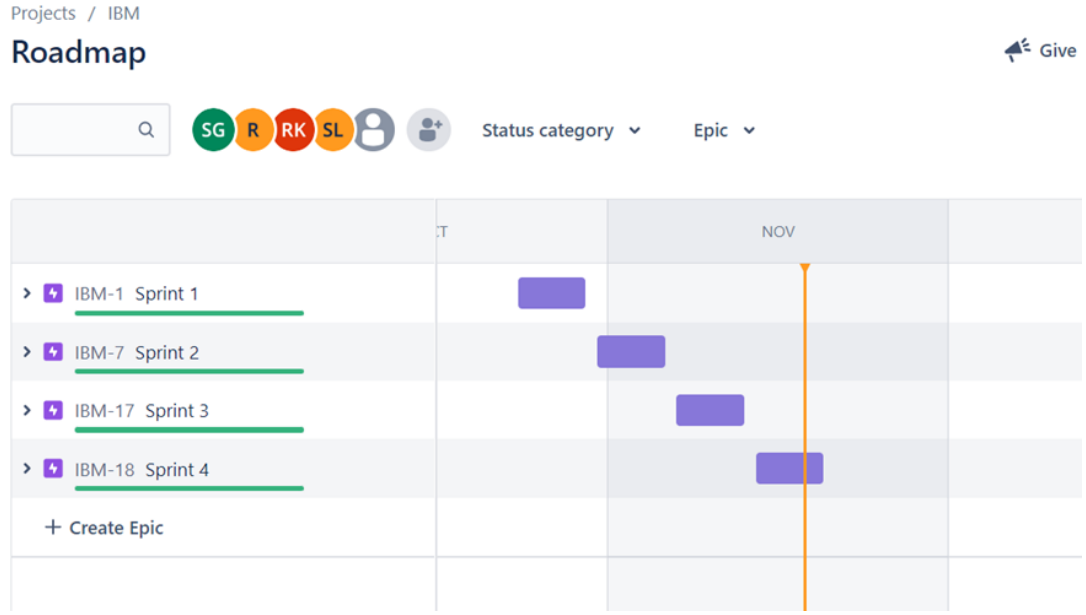
5	Data Collectionand Data pre- processing	Involves gathering and measuring information on variables to ensure accuracy and facilitate analysis. With preprocessing, necessary features were extracted for training the model.	1 WEEK	Completed
6	Model Building	Model Building is used for project visualization to deploy the system. The ouput will be a prediction of water quality index	1 WEEKS	In progress
7	Develop Application	A web application is application software that runsin a web browser, acts as an interface for the user top collect data and analyze the results.	2 WEEKS	In progress
8	Project Development Phase	Project development is theprocess of planning and allocating resources to fully develop aproject or product .	2 WEEKS	In progress

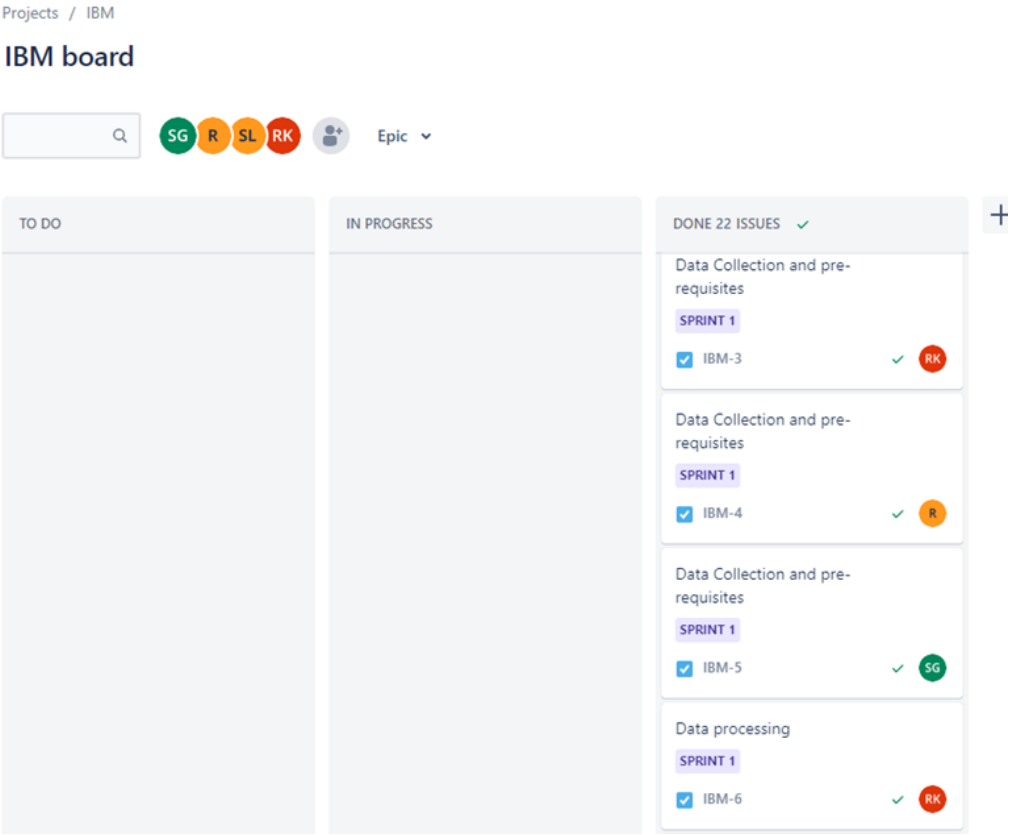
6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	Task	Story Points	Priority	Team Members
Sprint-1	Data preprocessing	USN-1	Processing the data prior to coding and making sure the dataset is usable for our application.	2	Medium	K Ramnath K Rohith Gupta
Sprint-1	Parameter inputs for prediction	USN-2	As a user, I can enter the test data required in the model for the quality prediction.	1	High	Santhosh Srinivas L K Ramnath
Sprint-2	Coding (Accessing and processing datasets)	USN-3	Coding is a set of instructions used to manipulate information so that a certain input results in a particular output.	2	High	G Sree Harine Santhosh Srinivas L K Ramnath
Sprint 3	Model Building and Evaluation	USN-4	Development of the model to perform testing to predict the water quality using the water quality index.	2	High	K Ramnath K Rohith Gupta G Sree Harine
Sprint-4	Web Application	USN-5	As a user, he/she will input the required parameters and obtain the prediction through the application.	1	Medium	Santhosh Srinivas L K Rohith Gupta

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

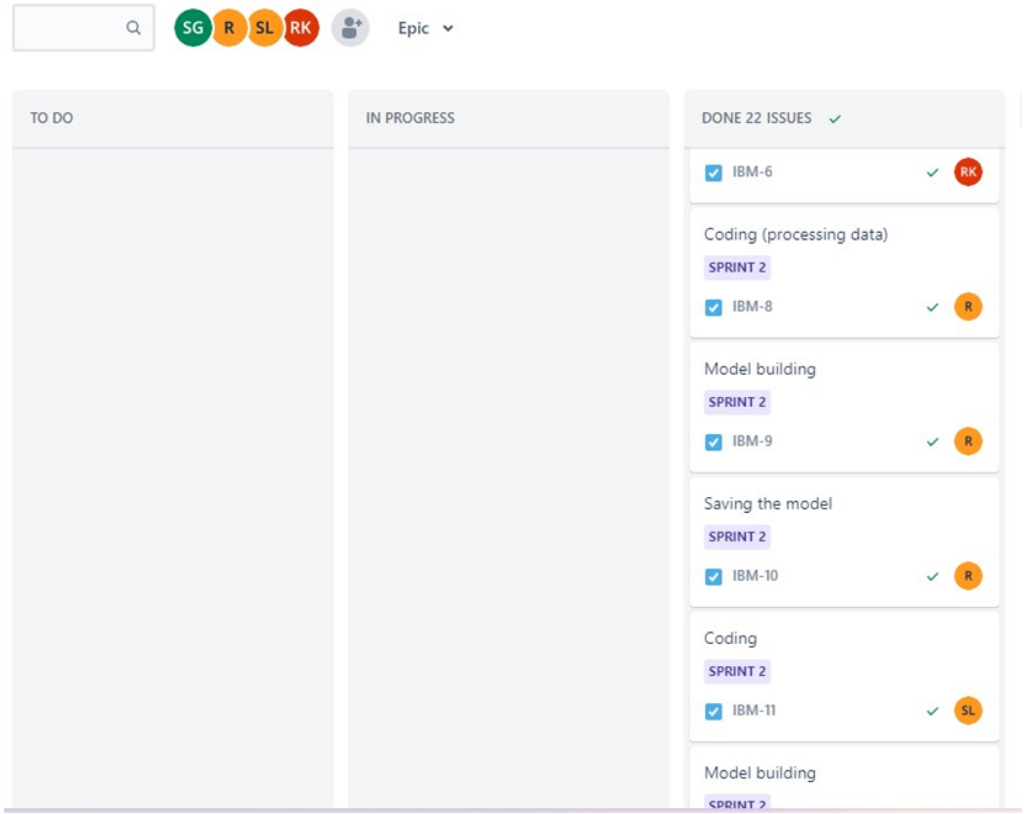
6.3 Reports from JIRA





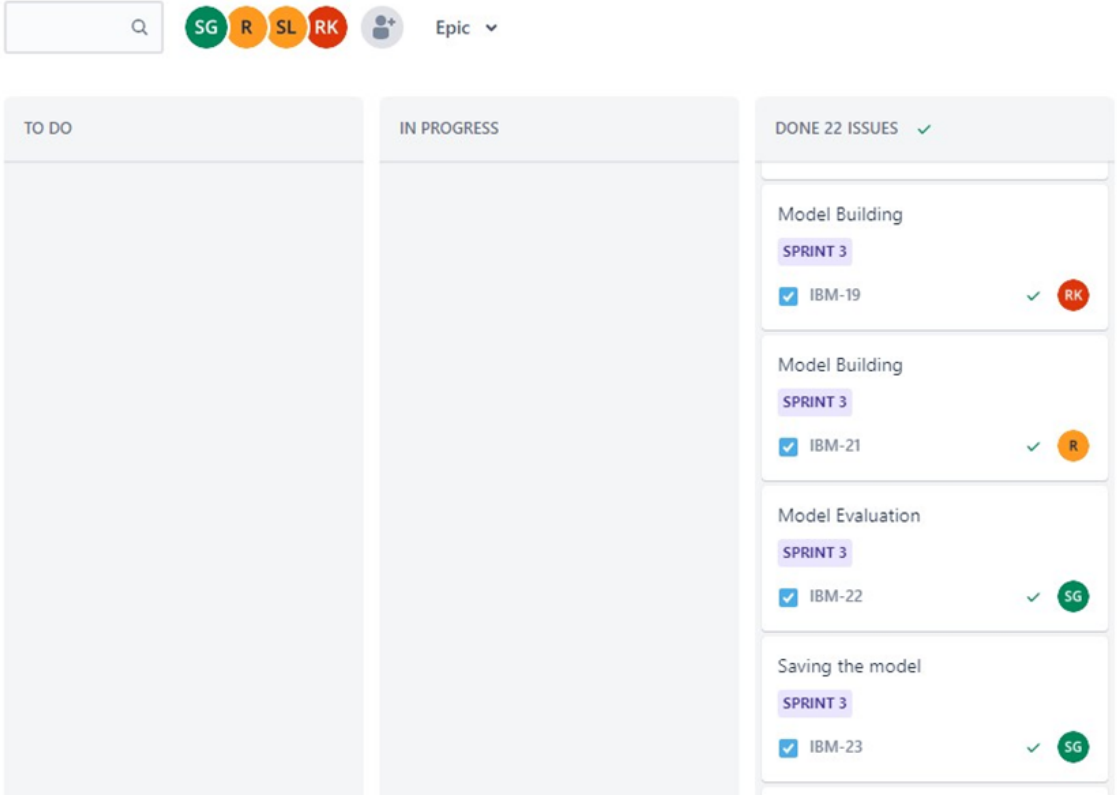
Projects / IBM

IBM board



Projects / IBM

IBM board



Projects / IBM

IBM board

SG R SL RK Epic

TO DO	IN PROGRESS	DONE 25 ISSUES
		Web application building SPRINT 4 IBM-24 SL
		Train Model in IBM Cloud SPRINT 4 IBM-25 RK
		Integrate Flask with Scoring end point SPRINT 4 IBM-26 R
		Registration on IBM Cloud SPRINT 4 IBM-27 SG
		Create login credentials page SPRINT 4 IBM-28 R
		User acceptance testing of the model SPRINT 4 IBM-29 RK
		Model verification on IBM SPRINT 4 IBM-30 SG

7. CODING & SOLUTIONING

7.1 Prediction Model

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
from sklearn.datasets import make_regression
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler
```

```
data=pd.read_csv("/content/water_dataX.csv",encoding='ISO-8859-1',low_memory=False)
data.head()
```

	STATION CODE	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATE NITRITEN (mg
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA	GOA	29.8	5.7	7.2	189	2	

```
data.describe()
```

	year
count	1991.000000
mean	2010.038172
std	3.057333
min	2003.000000
25%	2008.000000
50%	2011.000000
75%	2013.000000
max	2014.000000

```
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   STATION CODE                     1991 non-null   object
 1   LOCATIONS                        1991 non-null   object
 2   STATE                            1991 non-null   object
 3   Temp                             1991 non-null   object
 4   D.O. (mg/l)                      1991 non-null   object
 5   PH                               1991 non-null   object
 6   CONDUCTIVITY (µmhos/cm)          1991 non-null   object
 7   B.O.D. (mg/l)                    1991 non-null   object
 8   NITRATENAN N+ NITRITENANN (mg/l) 1991 non-null   object
 9   FECAL COLIFORM (MPN/100ml)       1991 non-null   object
 10  TOTAL COLIFORM (MPN/100ml)Mean   1991 non-null   object
 11  year                             1991 non-null   int64
dtypes: int64(1), object(11)
memory usage: 186.8+ KB
```

```
data.shape
```

```
(1991, 12)
```

```
data.isnull().any
```

```
<bound method NDFrame._add_numeric_operations.<locals>.any of       STATION CODE
LOCATIONS  STATE   Temp  D.O. (mg/l)     PH  \
0            False      False  False  False        False  False
1            False      False  False  False        False  False
2            False      False  False  False        False  False
3            False      False  False  False        False  False
4            False      False  False  False        False  False
...            ...        ...    ...    ...          ...    ...
1986         False      False  False  False        False  False
1987         False      False  False  False        False  False
1988         False      False  False  False        False  False
1989         False      False  False  False        False  False
1990         False      False  False  False        False  False

      CONDUCTIVITY (µmhos/cm)  B.O.D. (mg/l)  \
0                       False          False
1                       False          False
2                       False          False
3                       False          False
4                       False          False
...                       ...            ...
1986                    False          False
1987                    False          False
1988                    False          False
1989                    False          False
1990                    False          False

      NITRATENAN N+ NITRITENANN (mg/l)  FECAL COLIFORM (MPN/100ml)  \
0                                False                       False
1                                False                       False
2                                False                       False
3                                False                       False
4                                False                       False
...                                ...                         ...
1986                             False                       False
```

```
1987                                False                                False
1988                                False                                False
1989                                False                                False
1990                                False                                False

      TOTAL COLIFORM (MPN/100ml)Mean   year
0                                False  False
1                                False  False
2                                False  False
3                                False  False
4                                False  False
...                                ...    ...
1986                             False  False
1987                             False  False
1988                             False  False
1989                             False  False
1990                             False  False

[1991 rows x 12 columns]>
```

```
data.isnull().sum()
```

```
STATION CODE                        0
LOCATIONS                           0
STATE                               0
Temp                                0
D.O. (mg/l)                         0
PH                                  0
CONDUCTIVITY (µmhos/cm)             0
B.O.D. (mg/l)                       0
NITRATENAN N+ NITRITENANN (mg/l)    0
FECAL COLIFORM (MPN/100ml)          0
TOTAL COLIFORM (MPN/100ml)Mean      0
year                                0
dtype: int64
```

```
data.dtypes
```

```
STATION CODE                        object
LOCATIONS                           object
STATE                               object
Temp                                object
D.O. (mg/l)                         object
PH                                  object
CONDUCTIVITY (µmhos/cm)             object
B.O.D. (mg/l)                       object
NITRATENAN N+ NITRITENANN (mg/l)    object
FECAL COLIFORM (MPN/100ml)          object
TOTAL COLIFORM (MPN/100ml)Mean      object
year                                 int64
dtype: object
```

```
data["Temp"]=pd.to_numeric(data["Temp"],errors='coerce')
data["D.O. (mg/l)"]=pd.to_numeric(data["D.O. (mg/l)"],errors='coerce')
data["PH"]=pd.to_numeric(data["PH"],errors='coerce')
data["CONDUCTIVITY (µmhos/cm)"]=pd.to_numeric(data["CONDUCTIVITY (µmhos/cm)"],errors='coer
data["B.O.D. (mg/l)"]=pd.to_numeric(data["B.O.D. (mg/l)"],errors='coerce')
```

```
data["NITRATENAN N+ NITRITENANN (mg/l)"]=pd.to_numeric(data["NITRATENAN N+ NITRITENANN (mg
data["TOTAL COLIFORM (MPN/100ml)Mean"]=pd.to_numeric(data["TOTAL COLIFORM (MPN/100ml)Mean"
```

```
data.isnull().sum()
```

```
STATION CODE                         0
LOCATIONS                            0
STATE                                0
Temp                                92
D.O. (mg/l)                         31
PH                                   8
CONDUCTIVITY (µmhos/cm)             25
B.O.D. (mg/l)                       43
NITRATENAN N+ NITRITENANN (mg/l)    225
FECAL COLIFORM (MPN/100ml)           0
TOTAL COLIFORM (MPN/100ml)Mean      132
year                                 0
dtype: int64
```

```
data["Temp"].fillna(data["Temp"].mean(),inplace=True)
data["D.O. (mg/l)"].fillna(data["D.O. (mg/l)"].mean(),inplace=True)
data["PH"].fillna(data["PH"].mean(),inplace=True)
data["CONDUCTIVITY (µmhos/cm)"].fillna(data["CONDUCTIVITY (µmhos/cm)"].mean(),inplace=True
data["B.O.D. (mg/l)"].fillna(data["B.O.D. (mg/l)"].mean(),inplace=True)
data["NITRATENAN N+ NITRITENANN (mg/l)"].fillna(data["NITRATENAN N+ NITRITENANN (mg/l)"].m
data["TOTAL COLIFORM (MPN/100ml)Mean"].fillna(data["TOTAL COLIFORM (MPN/100ml)Mean"].mean(
data.isnull().any()
```

```
STATION CODE                        False
LOCATIONS                           False
STATE                               False
Temp                                False
D.O. (mg/l)                         False
PH                                  False
CONDUCTIVITY (µmhos/cm)             False
B.O.D. (mg/l)                       False
NITRATENAN N+ NITRITENANN (mg/l)    False
FECAL COLIFORM (MPN/100ml)          False
TOTAL COLIFORM (MPN/100ml)Mean      False
year                                False
dtype: bool
```

```
data.drop(["FECAL COLIFORM (MPN/100ml)"],axis=1,inplace=True)
```

```
data=data.rename(columns={'D.O. (mg/l)':'do'})
data=data.rename(columns={'PH':'ph'})
data=data.rename(columns={'CONDUCTIVITY (µmhos/cm)':'co'})
data=data.rename(columns={'B.O.D. (mg/l)':'bod'})
data=data.rename(columns={'NITRATENAN N+ NITRITENANN (mg/l)':'na'})
data=data.rename(columns={'TOTAL COLIFORM (MPN/100ml)Mean':'tc'})
data=data.rename(columns={'LOCATIONS':'location'})
data=data.rename(columns={'STATION CODE':'station'})
data=data.rename(columns={'STATE':'state'})
```

```
data.head()
```

	station	location	state	Temp	do	ph	co	bod
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179.0	1.700000
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64.0	3.800000

```
data['npH']=data.ph.apply(lambda x: (100 if (8.5>=x>=7)
                                  else(80 if (8.6>=x>=8.5) or (6.9>=x>=6.8)
                                       else(60 if (8.8>=x>=8.6) or (6.8>=x>=6.7)
                                           else(40 if (9>=x>=8.8) or (6.7>=x>=6.5)
                                                 else 0)))))


data['ndo']=data.do.apply(lambda x: (100 if (x>=6)
                                   else(80 if (6>=x>=5.1)
                                       else(60 if (5>=x>=4.1)
                                           else(40 if (4>=x>=3)
                                               else 0)))))

data['nco']=data.tc.apply(lambda x: (100 if (5>=x>=0)
                                          else(80 if (50>=x>=5)
                                            else(60 if (500>=x>=50)
                                              else(40 if (10000>=x>=500)
                                                else 0)))))


data['nbdo']=data.bod.apply(lambda x: (100 if (3>=x>=0)
                                          else(80 if (6>=x>=3)
                                            else(60 if (80>=x>=6)
                                              else(40 if (125>=x>=80)
                                                else 0)))))

data['nec']=data.co.apply(lambda x: (100 if (75>=x>=0)
                                          else(80 if (150>=x>=75)
                                            else(60 if (225>=x>=150)
                                              else(40 if (300>=x>=225)
                                                else 0)))))

data['nna']=data.na.apply(lambda x: (100 if (20>=x>=0)
                                          else(80 if (50>=x>=20)
                                            else(60 if (100>=x>=50)
                                              else(40 if (200>=x>=100)
                                                else 0)))))


data.head()
```

	station	location	state	Temp	do	ph	co	bod	na	tc	year
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049	0.1	27.0	2014
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000	0.2	8391.0	2014

```
data['wph']=data.npH*0.165
data['wdo']=data.ndo*0.281
data['wbdo']=data.nbdo*0.234
data['wec']=data.nec*0.009
data['wna']=data.nna*0.028
data['wco']=data.nco*0.281
data['wqi']=data.wph+data.wdo+data.wbdo+data.wec+data.wna+data.wco
data.head()
```

	station	location	state	Temp	do	ph	co	bod	na	tc	...	n
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049	0.1	27.0	...	
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000	0.2	8391.0	...	

```
average=data.groupby('year')['wqi'].mean()
average.head()
```

```
year
2003    66.239545
2004    61.290000
2005    73.762689
2006    72.909714
2007    74.233000
Name: wqi, dtype: float64
```

```
x=data.iloc[:,4:11].values
y=data.iloc[:,23:24].values
```

```
x.shape
```

```
(1991, 7)
```

```
y.shape
```

```
(1991, 1)
```

```
X_train, X_test, y_train,y_test = train_test_split(x,y,random_state=10,train_size=0.8)
```

```
X_test[0]
```

```
array([8.0000e-01, 6.8000e+00, 2.1560e+03, 7.4000e+00, 1.0000e+00,
       2.0733e+04, 2.0070e+03])
```

```
final_rf = RandomForestRegressor(n_estimators=100,random_state = 24).fit(X_train, y_train)
pred_final = final_rf.predict(X_test)
```

```
print('Mae:', metrics.mean_absolute_error(y_test,pred_final))
print('Mse:', metrics.mean_squared_error(y_test,pred_final))
print('Rmse:',np.sqrt(metrics.mean_absolute_error(y_test,pred_final)))
```

```
Mae: 0.8357609022556697
Mse: 4.872978975939852
Rmse: 0.914199596508153
```

```
metrics.r2_score(y_test,pred_final)
```

```
0.9734373994078458
```

7.2 Web Application

Html Code

```
index.html > ...
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>IBM - Water Analysis</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i5
    </head>
    <style>
        body {
          background-image: url('https://images.pexels.com/photos/261403/pexels-photo-261403.jpeg?cs=srgb&dl=pexels-pixabay-261403.jpg&fm=jpg');
          background-repeat: no-repeat;
          background-attachment: fixed;
          background-size: cover;
        }
        </style>
    <body>
    <!-- Navbar -->
        <nav class="navbar bg-light">
            <div class="container-fluid">
              <span class="navbar-brand mb-0 h1">Efficient Water Quality Analysis</span>
            </div>
        </nav>
        <div class="container" style="margin-top: 7.5%;">
            <div class="row">
                <div class="col-lg-8 col-12" style="margin-bottom: 2.5%;">
                    <p class="display-6" style="color: white;"><b>Enter the required parameters</b></p>
                    <form method="POST">
                        <div class="input-group mb-3">
                            <span class="input-group-text" id="param1">Year</span>
                            <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param1" >
                        </div>
                        <div class="input-group mb-3">
                            <span class="input-group-text" id="param2">pH</span>
                            <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param2">
```

```
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param2">
                </div>
                <div class="input-group mb-3">
                    <span class="input-group-text" id="param3">Conductivity (umhos/cm)</span>
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param3">
                </div>
                <div class="input-group mb-3">
                    <span class="input-group-text" id="param4">BOD (mg/L)</span>
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param4">
                </div>
                <div class="input-group mb-3">
                    <span class="input-group-text" id="param5">D.O (mg/L)</span>
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param5">
                </div>
                <div class="input-group mb-3">
                    <span class="input-group-text" id="param6">Nitritenann (mg/L)</span>
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param6">
                </div>
                <div class="input-group mb-3">
                    <span class="input-group-text" id="param7">Total coliform (mpn/100 ml)</span>
                    <input type="number" step="any" class="form-control" placeholder="Enter the value" name="param7">
                </div>
                <button type="submit" class="btn btn-primary">Predict</button>
            </form>
        </div>
        <div class="col-lg-4 col-12">
            <p class="display-6" style="color: white;"><b>Predicted Value</b></p>
            <ul class="list-group">
                <li class="list-group-item">Water Quality Index: {{result}}</li>
            </ul>
        </div>
    </div>

</div>
</body>
```

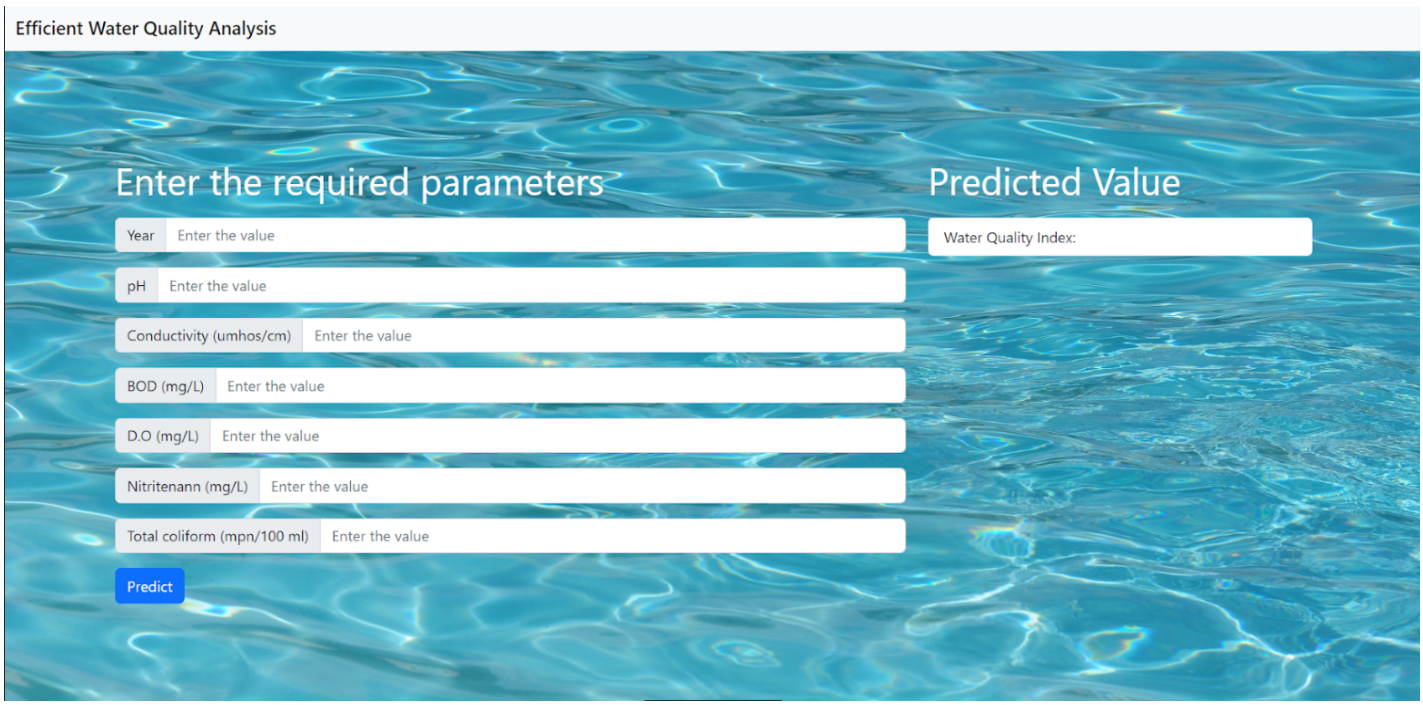
App.py

```python
from urllib import request
from flask import *
import joblib
import numpy as np

app = Flask(__name__)

@app.route('/',  methods=['GET', 'POST'])
def hello_world():
    res = ""
    if request.method == "POST":
        param1 = float(request.form.get("param1"))
        print(param1)
        param2 = float(request.form.get("param2"))
        print(param2)
        param3 = float(request.form.get("param3"))
        print(param3)
        param4 = float(request.form.get("param4"))
        print(param4)
        param5 = float(request.form.get("param5"))
        print(param5)
        param6 = float(request.form.get("param6"))
        print(param6)
        param7 = float(request.form.get("param7"))
        print(param7)
        loaded_rf = joblib.load("my_random_forest.joblib")
        test = [param5, param2, param3, param4, param6, param7, param1]
        X_test = np.array(test)
        print(X_test.shape)
        pred = loaded_rf.predict([X_test])
        print(pred)
        res = str(pred[0])
        return render_template("index.html", result=res)
    return render_template("index.html", result=res)
```

Webpage Screenshot



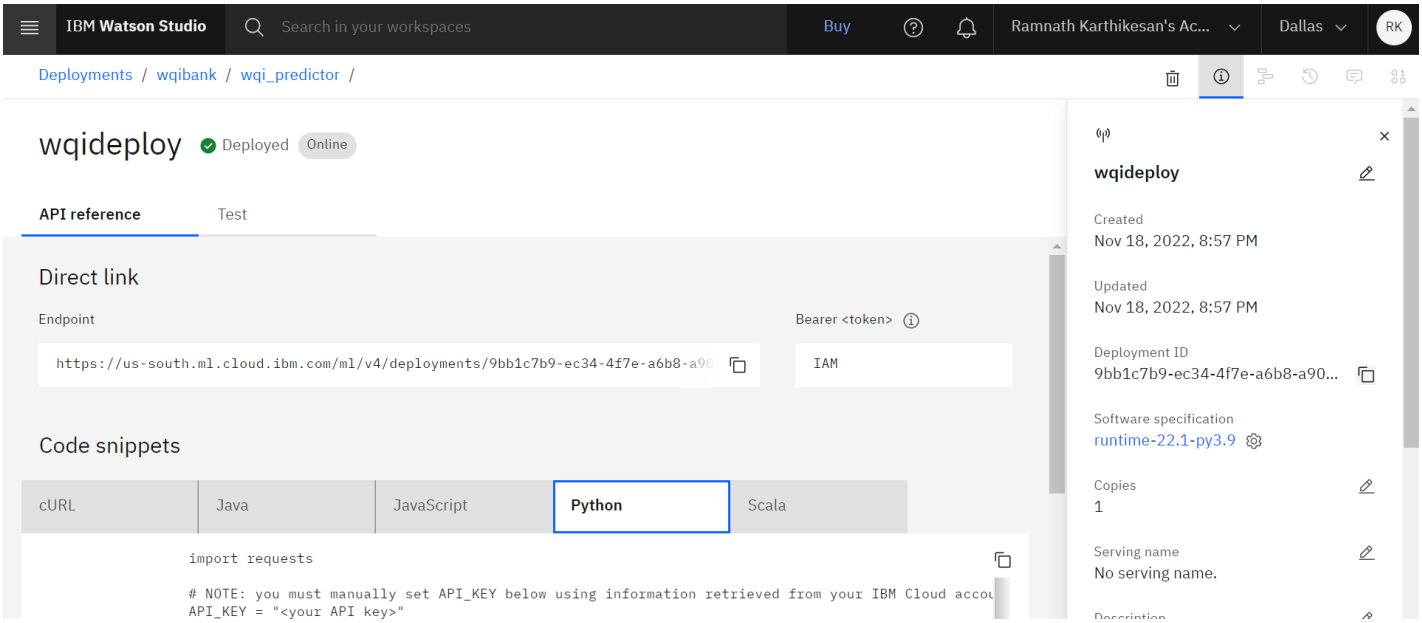
7.3 Database Schema

	STATION CODE	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATENAN N+ NITRITENANN (mg/l)	FECAL COLIFORM (MPN/100ml)	TOTAL COLIFORM (MPN/100ml)Mean	year
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	11	27	2014
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2	4953	8391	2014
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	3243	5330	2014
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	5382	8443	2014
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	3428	5500	2014
...	...	...	...	...	...	...	...	...	...	...	...	...
1986	1330	TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU	NAN	NAN	7.9	738	7.2	2.7	0.518	0.518	202	2003
1987	1450	PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, T...	NAN	29	7.5	585	6.3	2.6	0.155	0.155	315	2003
1988	1403	GUMTI AT U/S SOUTH TRIPURA,TRIPURA	NAN	28	7.6	98	6.2	1.2	NAN	NAN	570	2003
1989	1404	GUMTI AT D/S SOUTH TRIPURA, TRIPURA	NAN	28	7.7	91	6.5	1.3	NAN	NAN	562	2003
1990	1726	CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA	NAN	29	7.6	110	5.7	1.1	NAN	NAN	546	2003

1991 rows × 12 columns

7.4 IBM Watson Cloud Deployment

Model Deployed



Notebook Asset

In [12]:
```
rf = RandomForestRegressor(n_estimators=100,random_state=10).fit(X_train, Y_train)
```

In [22]:
```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "Dxm10MHxowMQ_iEVpafO4u_6KLcFX-1GJ-YyS6WW-5Ki"
}
client = APIClient(wml_credentials)
```

In [23]:
```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name'] == space_name)['metadata']['id'])
```

In [24]:
```
space_uid = guid_from_space_name(client, 'wqibank')
space_uid
```

Out[24]:
```
'fefcf0da-9870-42f5-a1fd-a0e557e7e5d7'
```

In [25]:
```
client.set.default_space(space_uid)
```

Out[25]:
```
'SUCCESS'
```

In [27]:
```
software_spec_uid = client.software_specifications.get_uid_by_name('runtime-22.1-py3.9')
software_spec_uid
```

Out[27]:
```
'12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

In [28]:
```
model_details = client.repository.store_model(model=rf, meta_props={
    client.repository.ModelMetaNames.NAME:"wqi_predictor",
    client.repository.ModelMetaNames.TYPE:"scikit-learn_1.0",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
})
```

In [30]:
```
model_id = client.repository.get_model_id(model_details)
```

In [ ]:
```

```

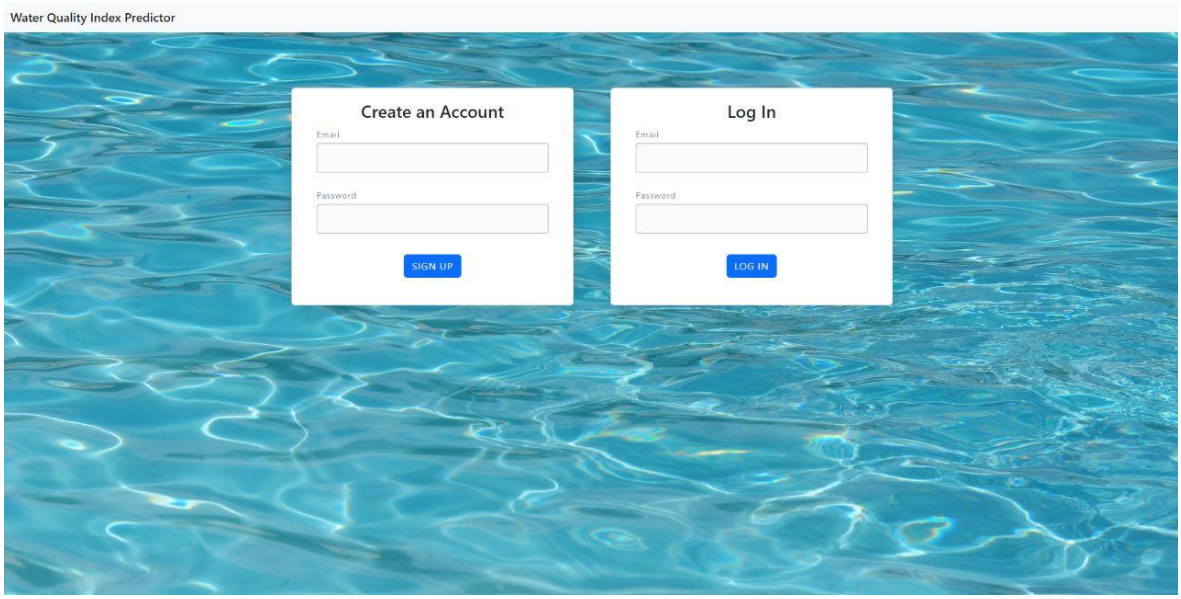
In [31]:
```
pred = rf.predict(X_test)
metrics.r2_score(Y_test,pred)
```

Out[31]:
```
0.9708947697439907
```

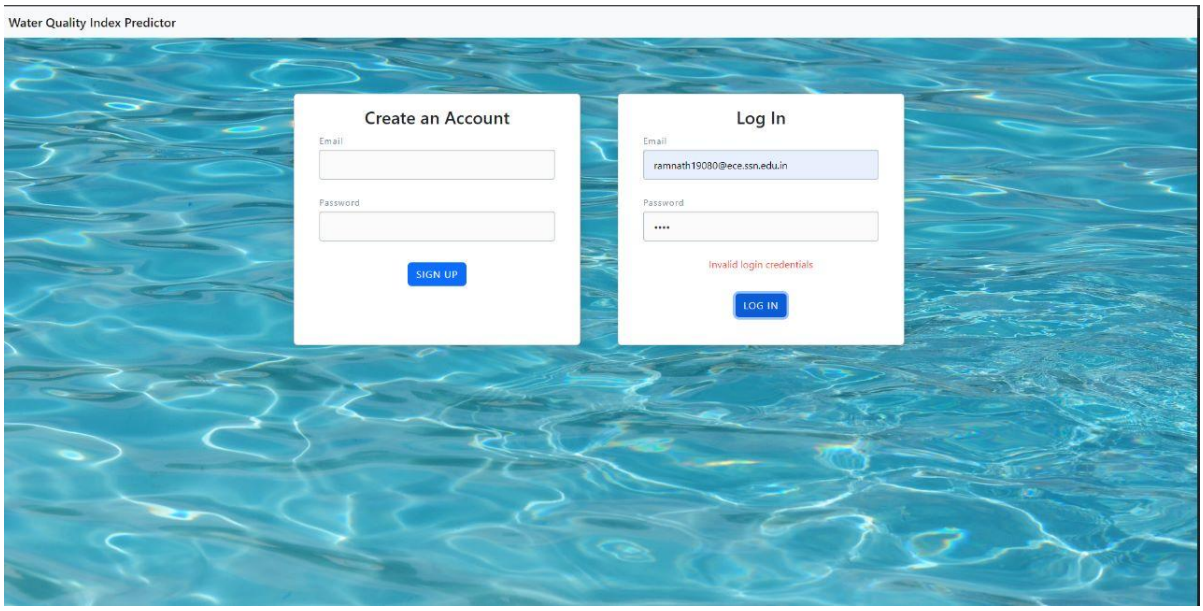
8. TESTING

8.1 Test Cases

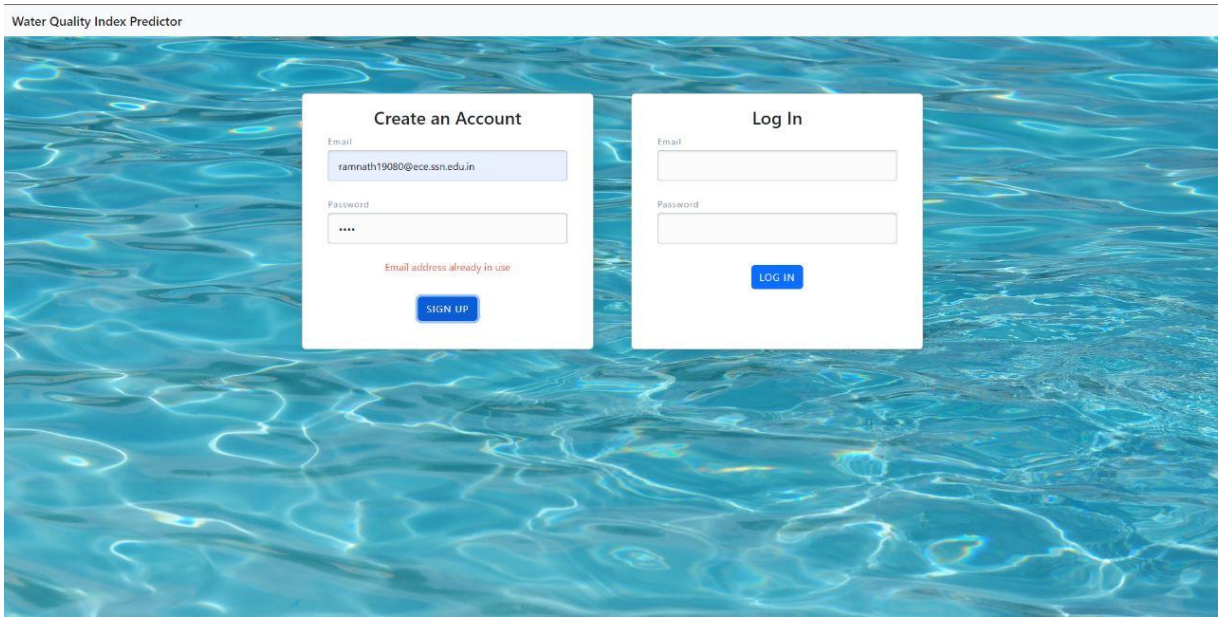
Login page (valid user):



Login with Invalid credentials (wrong email ID and password):



Existing registered user:



8.2 User Acceptance Testing

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	8	3	1	2	14
Duplicate	2	0	3	0	5
External	2	3	0	0	5
Fixed	11	2	5	20	38
Not Reproduced	0	0	1	0	1
Skipped	1	0	1	0	2
Won't Fix	0	5	0	0	5
Totals	24	13	11	22	70

Test case analysis (Total number of test cases = 399):

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	160	0	0	160
Client Application	150	0	0	150
Security	25	0	0	25
Outsource Shipping	24	0	0	24
Exception Reporting	16	0	0	16
Final Report Output	4	0	0	4
Version Control	20	0	0	20

9. RESULTS

9.1 Performance Metrics

Mean Absolute Error (MAE) = 0.8357609022556697

Mean Squared Error (MSE) = 4.872978975939852

Root Mean Squared Error(RMSE) = 0.914199596508153

Model Acccuracy (R-squared) = 0.9734373994078458

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Easy to implement
- Can be used in any location
- No expensive physical equipment required
- Prediction done using many parameters that improves the precision
- Faster results

Disadvantages:

- May not be accurate for all locations.
- Dynamic results can’t be obtained based on the present conditions.
- No physical measurement used to capture the location specific data.

11. CONCLUSION

This project has enabled us to create a web application that will analyze and predict the water quality efficiently through Machine learning techniques. With the utilization of the web application and a trained model, the prediction of water quality has been done using the water quality index calculation that tells us the measure of the water quality using several parameters in different locations. The quality index was efficiently calculated on the test data entered by the user.

12. FUTURE SCOPE

This model can be further developed to incorporate some more physical parameters (pollution measurements - for air, water and soil, proximity to industries etc.) to ensure greater accuracy of prediction specific to a particular location. Various other sensors can also be incorporated into this machine learning model to improve the location specific accuracy and make it into a dynamic model for testing the water quality. This can be made into a successful product for both domestic, agricultural and industrial applications.

13. APPENDIX

Source Code:

DataPrepocessing-https://colab.research.google.com/drive/18zTxGi4glnvfUT-z0Fp-7NydCrRvRVUj?usp=sharing

Model - https://colab.research.google.com/drive/1UfTeQ_k7kcLu-DNvkaw6M5szM713xjFz?usp=sharing

Final Web Application - https://github.com/Ramnath-Karthikesan/ibm-website/tree/master

Github IBM Project: https://github.com/Sreeharine/IBM-Project-20430-1659719139.git

Demo link: https://drive.google.com/file/d/1UYzmKlTj5O0iD64rvUmvE3PaRcUFutmk/view?usp=sharing