WQII APP

(WATER QUALITY INDEX INDIA)

A PROJECT REPORT

Submitted by

PRATYAKSH SHRIVASTAVA

(18BCE10195)

SUHANI AGRAWAL

(18BCE10272)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

VIT®
BHOPAL
www.vitbhopal.ac.in

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE

MADHYA PRADESH - 466114

APRIL 2022

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"WQII (WATER QUALITY INDEX INDIA)"** is the bonafide work of "**PRATYAKSH SHRIVASTAVA (18BCE10195) & SUHANI AGRAWAL (18BCE10272)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. Sandip Mal

Senior Assistant Professor
School of Computer Science & Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. Swagat Samantaray

Senior Assistant Professor
School of Computer Science & Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on **28/04/2022**

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr. Sandip Mal**, Head of the Department, School of Computer Science for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide **Dr. Swagat Samantaray**, for continually guiding and actively participating in my project, and giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night on the project to make it a success.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Water is an extremely important part of our lives. It is used for many purposes and in many ways. And for our environmental monitoring water quality testing holds a vital role.

Clean, safe, and adequate freshwater is of utmost importance to human existence and the survival of all living components in the ecosystem. Water quality issues are complex and diverse, deserving urgent global attention and action. The decline in water quality has become a global issue of concern because of its inherent ability to cause major alterations to the hydrological cycle. The past decade has seen a remarkable impact of man on the environment due to an unprecedented increase in population and rapid rate of urbanization as well as the intensification and expansion of agricultural practices. This has led to progressive and continual degradation of resources especially surface water. Polluted water is an important vehicle for the spread of diseases. In developing countries, about 1.8 million people, mostly children, die every year as a result of water-borne diseases.

The WATER QUALITY INDEX (WQI) provides a single number that expresses the overall water quality, at a certain location and time, based on several water quality parameters. Several indices have been developed to summarize water quality data in an easily expressible and easily understood format.

WQII (WATER QUALITY INDEX INDIA) is an application designed to determine the usability and quality of water in a particular region of India. It will check the water based on different parameters and then determine whether the water in this region is usable or not.

TABLE OF CONTENTS

CHAPTER – 1

PROJECT DESCRIPTION AND OUTLINE

1.1 INTRODUCTION

Clean, safe, and adequate freshwater is of utmost importance to human existence and the survival of all living components in the ecosystem. Water quality issues are complex and diverse, deserving urgent global attention and action. The decline in water quality has become a global issue of concern because of its inherent ability to cause major alterations to the hydrological cycle. The past decade has seen a remarkable impact of man on the environment due to an unprecedented increase in population and rapid rate of urbanization as well as the intensification and expansion of agricultural practices. This has led to progressive and continual degradation of resources especially surface water. Polluted water is an important vehicle for the spread of diseases.

In developing countries, about 1.8 million people, mostly children, die every year as a result of water-borne diseases.

Water quality is one of the most important factors in a healthy ecosystem. When water quality is poor, it affects not only aquatic life but also the surrounding ecosystem. Water testing is done based on different factors and parameters, it may be physical, chemical, or biological. It is especially important to understand and measure water quality as it directly impacts human consumption and health, industrial and domestic use, and the natural environment.

This application will check the water of the region on different parameters and will conclude that the water present here can be used for purposes like drinking, agriculture, washing, etc.

1. MOTIVATION FOR THE WORK

Poor water quality has a direct impact on water quantity in several ways. Polluted water that cannot be used for drinking, bathing, industry, or agriculture effectively reduces the amount of usable water within the given area. Maintaining good water quality is essential to human health, the environment, agricultural industries, and the recreational value of waterways, wetlands, and coastal waters. Awareness of the environmental conditions and human activities that influence water quality is important for effective water management. And it is very important to know whether the water in our area is usable or not and the major contents present in it. Our app will determine how and for what purpose we can use the water present in our region. So that it cannot cause any health-related issues. Or if the water present is contaminated, how can we make it usable.

1.3 PROBLEM STATEMENT

Due to the limited drinking water resources, intensive money requirements, growing population, urban change in rural areas, and the excessive use of sea resources for salt extraction have significantly worsened the water quality available to people. The high use of chemicals in manufacturing, construction, and other industries, and fertilizers in farms are also directly leaving the polluted water from industries into nearby water bodies have made a huge contribution to the global water quality reduction, which has become an important problem. Even due to containment water various water born are increasing day by day, due to which many human beings are losing their lives. Local water quality can be used to identify the sources and fates of toxic contaminants and pollutants either from ecology, geology, and anthropogenic activities (industrial processes, runoff from agricultural farms, etc) in the area.

1.4 OBJECTIVE OF THE WORK

The objective of WQI is to turn complex water quality data into information that is understandable and usable by the public. Several indices have been developed to summarize water quality data in an easily expressible and easily understood format. WQII is one of the most effective tools to express water quality and can be used as an important parameter for the assessment and management of the water source, giving a good idea of the evolutionary tendency of water quality to evolve over time.

- The index provides a single number that represents overall water quality at a certain location and time based on some water parameters.
- The water quality classification system used in the WQI denotes how suitable water is for drinking.
- The single-value output of this index, derived from several parameters, provides important information about water quality that is easily interpretable, even by laypeople.

CHAPTER 2

RELATED WORK INVESTIGATION

2.1 CORE AREA OF THE PROJECT

To determine the quality of a water body, the chemical, biological and physical conditions of a water body must be measured. Chemical measurements, biological surveys, and visual observations (physical) provide a "big picture" of what's happening in a water body. The following is a list of indicators (physical, chemical, and biological) that are often measured to assess the quality of water.

1. Physical indicators

Some physical indicators of the quality of a water sample from any source including,

- Temperature – Electrical Conductivity – Taste – Total Suspended Solids (TSS)
- Turbidity – Odour – Colour – Total Dissolved Solids (TDS)

2. Chemical indicators

Some chemical indicators of the quality of a water sample from any source including,

- pH – Biochemical Oxygen Demand (BOD) – Chemical Oxygen Demand (COD)
- Dissolved Oxygen (DO) – Total Hardness – Phosphates – Pesticides – Nitrates
- Surfactants – Heavy metals

Biological indicators

Some biological indicators of the quality of a water sample include,

- Bacteria (fecal coliform, Escherichia coli, Cryptosporidium, Giardia lamblia), − Viruses – Fungi protozoa – Parasitic worms)

2.2 EXISTING APPROACHES

2.2.1 APPROACH – 1

'Namami Gange Programme', is an Integrated Conservation Mission, approved as a 'Flagship Programme' by the Union Government in June 2014 with a budget outlay of Rs.20,000 Crore to accomplish the twin objectives of effective abatement of pollution, conservation, and rejuvenation of National River Ganga.The aims and objectives of NMCG are to accomplish the mandate of the National Ganga River Basin Authority (NGRBA) of.
1. To ensure effective abatement of pollution and rejuvenation of the river Ganga by adopting a river basin approach to promote inter-sectoral co-ordination for comprehensive planning and management.
2. To maintain minimum ecological flows in the river Ganga to ensure water quality and environmentally sustainable development.

Figure 2.2.1.1 Namami Gange

2.2.3 APPROACH – 2

In Nigeria, the most frequent water sources are surface waters (rivers, streams, ponds, and lakes) and groundwater (borehole and hand-dug wells). The Physico-chemical assessments of water samples showed that while some of the parameters are within permissible limits, many exceeded the stipulated standards. Application of the water quality index (WQI) to determine the suitability of the water for an intended use indicated that most water sources in the western part of the country are good and suitable for human consumption except for incidences of high levels of fecal contamination in some rivers. The WQI for most locations in the northern part of the country is either bad or poor and not suitable for human consumption. In the eastern and southern parts of the country, the WQI index indicated marginal quality that was not suitable for human consumption without treatment. This marginal quality could be a result of the high levels of nitrate and acidic pH of most of the water bodies in the area. In all, it is recommended that prior treatment of the water is very important before consumption to avoid water-borne related diseases and illnesses.

2.2.3 APPROACH – 3

The Delphi technique was used for selecting water quality parameters in several WQI model applications. Here, the important parameters are selected based on gathering expert opinions through interviews or surveys. In general, there are no specific rules or guidelines for selecting the water quality parameter for inclusion in the WQI model. The traditional WQI model does not follow any systematic technique for setting its parameters. It seems that the WQI model parameters were generally chosen based on a few common water quality issues such as oxygen availability, eutrophication, health considerations, physical and chemical phenomena, and dissolved constituents.

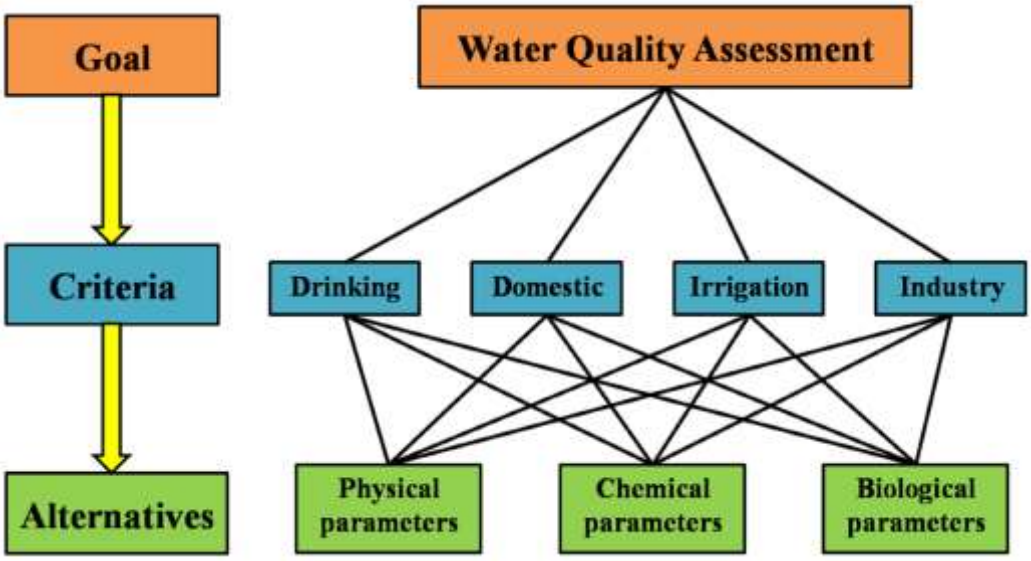

Figure 2.2.3.1 Delphi Technique

2.3 ISSUES

1. Lack of precision and accuracy in classification technique of importance of evaluation of parameters.
2. Inefficiency in dealing with uncertainty and subjectivity in a complex environmental issue such as the incompatibility of observations, uncertainty, and imprecision in criteria.
3. Lack of a uniform method for measuring water pollution involving biological parameters.
4. Inadequate transfer of complex environmental data into information.
5. Manual data entry and calculation required.
6. No Live data.

CHAPTER – 3

REQUIREMENT ARTIFACTS

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE:

We are assuming that we already have the hardware that are the devices installed by the government of India on the Ganga River. If we want to rely on actual information and not on data. We need hardware for every water body to see whether the water can be useful or not.

SOFTWARE:

- An Android Phone with Android 6.0 or above.
- The basic dataset & Information and calculation.
- 150 Mb of free space.

3.2 SPECIFIC PROJECT REQUIREMENTS

We have the data available with us saying how the water of different regions differs based on different parameters.

The project "NAMAMI GANGE'' has all the data sets of the Ganga River at different places and regions based on different parameters.

Data presented in the Freshwater Quality Index helps indicate whether water quality is good, meeting standards to protect aquatic life, whether it is of moderate concern or is poor, and doesn't meet expectations. The index ranges from 1 to 100; a higher number indicates better water quality.

We took that data and worked on it and use it for our application WQII.

Table 3.2.1 Usable water quality table

PARAMETERS		MIN	MAX
TEMPERATURE(`C)	FOR BATHING	32	40
	FOR DRINKING	6	20
	FOR AGRICULTURE	-	
	FOR WASHING	32	95
Dissolved Oxygen (mg/L)	FOR DRINKING	6.5	8
	FOR AGRICULTURE	5	10
pH	FOR DRINKING	6.5	8.5
	FOR BATHING	5.5	6
	FOR WASHING	8	10.5
	FOR AGRICULTURE	5	8
	FOR DRINKING	200	800
	FOR AGRICULTURE	1500	2500
Bio- Chemical Oxygen Demand(mg/L)	FOR DRINKING	0	1
	FOR WASHING	0	5
	FOR AGRICULTURE	0	200
Nitrate(mg/L)	FOR DRINKING	0	10
	FOR BATHING	10	
	FOR WASHING	10	
	FOR AGRICULTURE		50
Faceal Coliform(MPN/100mL)	FOR BATHING		500
	FOR DRINKING		0
	FOR AGRICULTURE		1000

Table 3.2.2 Water quality of river ganga

Station Code	LOCATION	STATE	Temperature (°C)		Dissolved Oxygen (mg/ L)		pH		Conductivity (µmho/ Cm)		Bio-Chemical Oxygen Demand (mg/L)		Nitrate (mg/ L)		Faecal Coliform (MPN/ 100 mL)		Total Coliform (MPN/ 100 mL)		Faecal Streptococci (MPN/ 100 mL)	
					> 5.0 mg/L		6.5 - 8.5				< 3.0 mg/L				< 2500 MPN/ 100 mL					
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
1491	BHAGIRATHI AT GANGOTRI	UTTARAKHAND	-	-	9.8	9.8	-	-	-	-	-	-	-	-	7	7	9	9	-	-
1481	MANDAKINI B/C ALAKNADA AT RUDRAPRAYAG,	UTTARAKHAND	17	18	9.4	9.8	7.8	7.9	-	-	BDL	1.0	-	-	1.8	2	1.8	2	-	-
1484	ALAKNANDA B/C MANDAKINI AT RUDRA PRAYAG	UTTARAKHAND	17	19	9.6	10.2	7.6	7.8	-	-	1.0	1.0	-	-	1.8	2	NA	2	-	-
1486	ALAKNANDA A/C MANDAKINI AT RUDRAPRAYAG	UTTARAKHAND	16	18	9.2	10.2	7.7	7.8	-	-	1.0	1.0	-	-	1.8	2	NA	2	-	-
1488	BHAGIRATHI B/C WITH ALAKNANDA AT DEVPRAYAG	UTTARAKHAND	17	18	9.4	10.1	7.6	8.0	-	-	1.0	1.0	-	-	1.8	2	NA	2	-	-
1487	ALAKNANDA B/C TO BHAGIRATHI AT DEVPRAYAG	UTTARAKHAND	17	19	9.2	10.0	7.6	7.9	-	-	1.0	1.0	-	-	1.8	2	NA	2	-	-
1489	ALAKNANDA A/C WITH BHAGIRATHI AT DEVPRAYAG	UTTARAKHAND	17	19	9.6	10.0	7.9	8.0	-	-	1.0	1.0	-	-	1.8	2	NA	2	-	-
20048	SWARG ASHRAM-1	UTTARAKHAND	19	21	9.2	10.2	7.9	8.1	105	293	1.0	1.0	-	-	40	60	110	170	-	-
1060	GANGA AT RISHIKESH U/S	UTTARAKHAND	16	20	9.8	11.8	7.4	8.2	114	125	1.0	1.0	__	__	14	30	40	401	-	-
20049	LAKKAR GHAT- OXIDATION PONDS	UTTARAKHAND	19	28	8.8	10.1	7.8	8.4	-	390	1.0	1.0			40	70	90	170	-	-
10147	RIVER GANGA D/S RISHIKESH	UTTARAKHAND	16	22	9.0	11.2	7.3	8.2	97	154	1.0	1.0	__	__	26	40	50	90	-	-
2725	GANGA RIVER AFTER CONFLUENCE OF RIVER SONG NEAR SATYANARAYAN TEMPLE D/S RAIWALA, DEHRADUN	UTTARAKHAND	16	25	8.6	10.4	7.4	8.4	308	400	1.0	1.2	__	__	50	130	110	280	-	-
10148	RIVER GANGA AT HAR KI PAURI GHAT	UTTARAKHAND	17	22	9.0	10.2	7.5	8.4	121	155	1.0	1.0	__	__	26	50	40	110	-	-
1061	GANGA AT HARIDWAR D/S, UPPER GANGA CANAL D/S BALKUMARI MANDIR,	UTTARAKHAND	16	22	8.6	9.4	7.1	8.0	173	231	1.0	2.0	__	__	70	220	130	350	-	-
2727	UPPER GANGA RIVER D/S ROORKEE	UTTARAKHAND	16	20	8.6	10.0	7.4	8.3	-	163	1.0	1.2	-	-	40	90	110	170	-	-
10150	RIVER GANGA AT MADHYA GANGA BARRAGE, BIJNOR	UTTAR PRADESH	__	__	7.5	9.7	7.4	7.7	62	253	1.3	1.8	__	__	__	__	__	__	-	-
1062	GANGA AT GARHMUKTESHWAR	UTTAR PRADESH	16.5	26	7.5	10.5	6.8	7.7	155	266	0.9	2.8	0.33	0.82	220	1700	540	3500	-	-
10149	GHARMUKTESHWAR	UTTAR PRADESH	16	26	7.3	11.5	6.5	7.6	163	256	0.6	2.1	0.25	0.85	110	1300	240	2800	-	-
2488	GANGA U/S, ANOOPSHAHAR	UTTAR PRADESH	15	30	7.0	10.4	6.9	7.5	210	232	1.1	2.8	0.32	0.62	110	280	170	350	-	-
2489	GANGA D/S, ANOOPSHAHAR	UTTAR PRADESH	17	30	7.2	10.3	7.0	7.6	218	236	0.8	2.6	0.34	0.53	170	430	240	920	-	-
1145	GANGA AT NARORA (BULANDSAHAR)	UTTAR PRADESH	14	29	7.3	10.6	7.1	7.6	220	234	1.0	2.5	0.37	0.66	280	920	630	1600	-	-
2490	GANGA AT KACHHLA GHAT, ALIGARH	UTTAR PRADESH	16	25	9.7	10.5	7.0	8.0	118	789	0.9	2.5	0.55	0.62	170	270	250	460	-	-
10151	RIVER GANGA AT FARUKKABAD	UTTAR PRADESH	16	33	6.4	10.7	7.6	8.4	157	285	1.7	2.6	__	__	1.8	1700	1700	2700	-	-
1063	GANGA AT KANNAUJ U/S (RAGHAT)	UTTAR PRADESH	16	32	6.4	10.6	7.7	8.6	205	345	2.4	3.4	__	__	1300	2700	3300	4600	-	-
1066	GANGA AT KANNAUJ D/S	UTTAR PRADESH	16	32	6.0	10.2	7.8	8.6	200	366	2.8	4.2	__	__	1700	3300	4000	5800	-	-
1146	GANGA AT BITHOOR (KANPUR)	UTTAR PRADESH	16	32	6.1	10.6	7.9	8.8	203	350	2.3	4.4	__	__	1400	3400	3300	5800	-	-
1067	GANGA AT KANPUR U/S (RANIGHAT)	UTTAR PRADESH	16	32	5.8	10.3	7.8	8.7	203	368	2.5	4.0	__	__	1700	3400	3300	5800	-	-
10154	RIVER GANGA AT BATHING GHAT (BHARADGHAT)	UTTAR PRADESH	16	32	5.7	10.4	7.8	8.7	206	362	2.4	3.8	__	__	2000	3300	3200	4900	-	-
10155	RIVER GANGA AT D/S SHUKLAGANJ	UTTAR PRADESH	16	32	5.4	10.0	7.8	8.7	210	380	2.7	4.4	__	__	2000	4300	4600	7000	-	-
1068	GANGA AT KANPUR D/S (JAJMAU PUMPING STATION)	UTTAR PRADESH	16	32	4.6	9.4	7.6	8.4	223	442	3.2	5.8	__	__	17000	35000	28000	63000	-	-
10157	RIVER GANGA AT BATHING GHAT (JAJMAU BRIDGE)	UTTAR PRADESH	16	32	5.6	9.6	7.7	8.5	219	392	2.8	5.4	__	__	4300	25000	7000	54000	-	-
1147	GANGA AT DALMAU (RAI BAREILLY)	UTTAR PRADESH	17	27	7.0	10.6	7.9	8.7	201	458	2.8	4.1	0.78	0.86	2500	3600	3100	4100	-	-
2498	GANGA AT KALA KANKAR, RAEBARELI	UTTAR PRADESH	17	27	6.9	11.0	7.9	8.5	206	456	2.7	4.1	0.76	0.84	2400	3400	3100	4000	-	-
1066	GANGA AT ALLAHABAD (RASOOLABAD)	UTTAR PRADESH	22.3	32.5	7.3	12.2	7.6	8.4	154	261	2.1	3.2	1.28	1.61	4300	14000	9400	24000	-	-
2487	GANGA AT KADAGHAT, ALLAHABAD	UTTAR PRADESH	22.4	32.7	6.9	11.5	7.8	8.5	150	268	2.0	3.9	1.38	1.77	4600	14000	11000	27000	-	-
1048	GANGA AT ALLAHABAD D/S (SANGAM)	UTTAR PRADESH	22.6	32.8	6.8	11.5	7.8	8.4	236	383	2.0	3.4	1.39	1.83	3300	13000	8400	27000	-	-
10158	RIVER GANGA A/C TAMSA RIVER, SIRSA, SON BARSA	UTTAR PRADESH	22.7	32.8	7.0	11.7	7.7	8.2	346	376	1.8	2.9	1.35	1.69	3100	9400	7000	22000	-	-
2485	GANGA U/S, VINDHYACHAL, MIRZAPUR	UTTAR PRADESH	19	29.5	7.2	9.2	8.2	8.4	352	468	2.1	3.2	0.48	0.90	800	1300	1300	2700	-	-
2486	GANGA D/S, MIRZAPUR	UTTAR PRADESH	19	29.8	6.7	8.4	8.1	8.6	364	496	3.3	3.9	0.84	1.30	7000	11000	13000	22000	-	-
10159	RIVER GNAGA TA CHUNAR	UTTAR PRADESH	19	29.5	6.9	8.8	2.2	8.5	360	488	2.9	3.8	0.77	1.12	5000	8000	8000	14000	-	-

Station Code	LOCATION	STATE	Temperature (° C)		Dissolved Oxygen (mg/ L)		pH		Conductivity (µmho/ Cm)		Bio-Chemical Oxygen Demand (mg/L)		Nitrate (mg/ L)		Faecal Coliform (MPN/ 100 mL)		Total Coliform (MPN/ 100 mL)		Faecal Streptococci (MPN/ 100 mL)	
					> 5.0 mg/L		6.5 - 8.5				< 3.0 mg/L				< 2500 MPN/ 100 mL					
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
1070	GANGA AT VARANASI U/S (ASSIGHAT)	UTTAR PRADESH	20	31.5	7.2	10.0	8.1	8.4	364	513	1.7	3.3	0.12	0.24	500	1700	1100	3400	-	-
1071	GANGA AT VARANASI D/S (MALVIYA BRIDGE)	UTTAR PRADESH	20	31.5	6.5	7.9	7.9	8.7	392	528	3.4	4.2	0.15	0.40	11000	23000	17000	43000	-	-
10151	RIVER GANGA A/C GOMTI RIVER, BHUSAULA	UTTAR PRADESH	20.5	32.5	6.8	8.0	8.2	8.6	370	524	3.0	3.8	0.15	0.31	5000	11000	8000	17000	-	-
10156	RIVER GANGA AT BATHING GHAT (GOLA GHAT)	UTTAR PRADESH	16	32	5.6	10.1	7.7	8.6	236	384	2.6	4.6	-	-	2300	5400	4600	13000	-	-
1073	GANGA AT TRIGHAT (GHAZIPUR)	UTTAR PRADESH	20.5	32	6.5	7.8	8.1	8.7	388	535	3.3	4.3	0.19	0.38	11000	17000	21000	31000	-	-
1074	GANGA AT BUXAR	BIHAR	17	35	6.8	10.2	7.0	9.0	250	512	1.4	2.8	0.02	0.37	450	28000	2300	160000	-	-
10113	RIVER GANGA AT U/S JAIL GHAT, BUXAR	BIHAR	17	37	6.0	8.6	7.1	9.6	253	508	1.5	2.7	0.03	0.46	2100	29000	9400	160000	-	-
2551	GANGA AT BUXAR, RAMREKHAGHAT	BIHAR	17	36	6.0	8.3	7.3	8.7	257	523	1.9	2.8	0.02	0.54	1100	17000	9400	35000	-	-
1113	GANGA AT D/S BUXAR NEAR ROAD BRIDGE	BIHAR	16	34	6.3	8.5	7.2	8.8	248	629	1.5	2.7	0.02	0.51	170	17000	1600	54000	-	-
10131	RIVE GANGA AT U/S BARAHPUR BINDTOLI	BIHAR	19	32	6.2	9.3	7.5	8.8	236	411	1.3	2.2	0.14	0.73	1700	17000	11000	92000	-	-
10162	GANGA AT AAA-CHHAPRA ROAD BRIDGE (KOILWER BANJRA- CHHAPRA ROAD)	BIHAR	18	35	5.7	8.8	7.3	8.3	250	505	1.4	2.0	0.02	0.36	1300	3000	3300	35000	-	-
2564	GANGA AT THE CONFLUENCE OF SONE RIVER DORIGANJ, CHAPRA	BIHAR	17	33	7.3	9.8	7.3	8.1	237	457	1.4	2.3	0.10	0.21	1400	4000	2600	28000	-	-
10114	RIVER GANGA TA MAA AMBIKA ASTHAN, AAMI, SARAN	BIHAR	17	32	7.2	8.7	7.2	8.0	16	462	1.7	2.2	0.12	0.62	450	22000	14000	43000	-	-
3114	GANGA NEAR DANAPUR (NEAR PIPA PUL) ,PATNA	BIHAR	19	33	6.7	9.4	7.4	8.5	283	563	1.3	2.4	0.01	0.86	780	14000	3100	28000	-	-
1077	GANGA AT KURJI, PATNA U/S	BIHAR	18	33	6.3	9.0	7.4	8.5	257	563	1.3	2.0	0.01	1.00	2100	11000	2600	35000	-	-
2552	GANGA AT GANDHIGHAT, NIT AT PATNA	BIHAR	19	33	6.5	8.8	7.7	8.3	283	467	1.8	2.9	0.03	0.58	1400	26000	11000	35000	-	-
10115	RIVER GANGA TA GULABI GHAT, PATNA	BIHAR	19	34	5.6	8.3	7.4	8.4	263	465	1.8	2.8	0.03	0.82	1400	22000	6800	92000	-	-
1079	GANGA AT PATNA D/S (GANGA BRIDGE), GAIGHAT	BIHAR	19	34	6.3	8.8	7.7	8.6	250	479	1.7	2.6	0.11	1.00	1100	29000	11000	160000	-	-
3122	GANGA AT MALSALAMI, PATNA	BIHAR	18	33	7.0	9.3	7.3	8.5	247	458	1.5	2.6	0.02	0.80	1100	11000	4000	24000	-	-
4297	KADHCHI-DARGAH-BIDUPUR ROAD BRIDGE, PATNA	BIHAR	18	34	6.6	9.8	6.6	8.5	262	456	1.2	2.1	0.02	0.78	1100	11000	5500	28000	-	-
10122	RIVER GANGA AT TRIVENI GHAT, PATNA	BIHAR	17	33	6.1	9.5	7.7	8.4	245	454	1.4	2.7	0.03	0.64	1400	21000	3300	28000	-	-
2553	GANGA AT KEWALA GHAT,FATUHA, PATNA	BIHAR	17	34	6.2	8.5	7.5	8.2	252	467	1.8	2.9	0.02	1.83	1100	22000	2800	92000	-	-
4301	BAKHTHARPUR TAJPUR BRIDGE ON GANGA, ATHMALGOLA, PATNA	BIHAR	18	33	5.8	9.4	7.4	8.6	246	514	1.3	2.2	0.09	0.61	1700	17000	3100	54000	-	-
3115	GANGA RIVER AT BARH	BIHAR	19	33	6.1	9.8	7.5	9.9	238	462	1.3	2.2	0.03	0.62	1300	22000	7000	54000	-	-
10130	RIVER GANGA AT D/S NAWADIGHAT, BARH, PATNA	BIHAR	18	32	6.0	9.6	7.5	8.9	240	475	1.1	2.3	0.10	0.58	1400	17000	1900	160000	-	-
1017	GANGA AT MOKAMA (U/S), MAHADEO ASTHAN, PATNA	BIHAR	18	33	6.1	9.4	7.4	8.8	257	452	1.1	2.6	0.14	0.90	1100	13000	1400	54000	-	-
1015	GANGA AT MOKAMA (D/S)	BIHAR	19	33	6.3	8.0	7.2	8.5	208	425	1.6	2.6	0.06	0.54	680	17000	14000	35000	-	-
3123	GANGA RIVER AT BARAHIYA, NEPALITOLA, MAMANCHI, LAKHISARAI	BIHAR	21	35	6.1	9.2	7.6	8.7	239	452	1.1	2.3	0.11	1.17	1100	14000	11000	35000	-	-
3116	GANGA AT U/S MUNGER (NEAR INTAKE WELL ON GANGA)	BIHAR	20	32	6.0	9.7	7.6	8.6	232	440	1.1	2.5	0.03	0.85	1100	17000	11000	160000	-	-
1018	GANGA AT MUNGER	BIHAR	21	32	5.8	9.6	7.6	8.5	234	436	1.5	2.5	0.05	0.60	1100	28000	14000	160000	-	-
3117	GANGA AT U/S SULTANGANJ, BHAGALPUR	BIHAR	18	32	6.1	9.5	7.4	8.7	243	427	1.5	2.3	0.07	0.50	1700	17000	9300	92000	-	-
4399	GANGA RIVER AT WATER INTAKE POINT, BHAGALPUR	BIHAR	18	34	6.1	8.7	7.3	9.1	239	512	1.3	2.6	0.05	1.06	1200	22000	4100	92000	-	-
2554	GANGA AT SULTANGANJ, BHAGALPUR	BIHAR	18	32	6.2	9.4	7.3	8.6	234	408	1.3	2.6	0.04	0.56	780	28000	3900	160000	-	-
10138	RIVER GANGA AT CHAMPANAGAR, BHAGALPUR	BIHAR	17	35	6.5	8.7	7.0	8.6	241	518	1.5	2.6	0.06	0.65	200	28000	2700	160000	-	-
1019	GANGA AT BHAGALPUR	BIHAR	18	32	6.1	8.6	7.6	8.3	224	505	1.4	2.5	0.01	0.91	780	17000	6800	92000	-	-
3118	GANGA AT U/S BHAGALPUR NEAR BARARIGHAT	BIHAR	13	34	6.0	8.5	7.6	8.6	244	500	1.9	2.7	0.01	0.52	930	35000	11000	160000	-	-
1016	GANGA AT KAHALGAON	BIHAR	18	32	6.1	9.5	7.5	8.6	221	629	1.1	2.3	0.07	0.56	200	17000	11000	160000	-	-
10143	RIVER GANGA D/S NEAR CREMATION GHAT	BIHAR	18	32	5.9	9.3	7.4	8.5	236	426	1.2	2.6	0.01	0.65	1400	14000	14000	160000	-	-
10144	RIVER GANGA U/S NEAR ICT GHAT	JHARKHAND	17	40	7.8	8.6	7.8	8.4	304	316	2.2	2.6	—	—	—	—	—	—	-	-
10145	RIVER GANGA D/S NEAR JANTA GHAT	JHARKHAND	18	44	8.0	8.6	7.8	8.4	312	320	2.2	2.6	—	—	—	—	—	—	-	-
10146	RIVER GANGA AT SAHIBGANJ	JHARKHAND	19	39	8.2	8.6	8.0	8.3	312	318	2.2	2.8	—	—	—	—	—	—	-	-
1059	GANGA AT RAJMAHAL	JHARKHAND	18	40	8.0	8.6	8.1	8.6	310	322	2.2	2.8	—	—	—	—	—	—	-	-
1080	GANGA AT BAHARAMPORE, MURSHIDABAD	WEST BENGAL	17	37	4.9	11.2	7.4	8.5	198	380	0.8	4.8	0.28	0.73	7000	300000	14000	500000	-	-
10159	RIVER GANGA AT BEHRAMPORE (KHAGRA) MURSHIDABAD	WEST BENGAL	17	37	4.8	11.0	7.2	8.5	193	367	0.9	5.8	0.29	0.82	1300	280000	13000	350000	-	-
10160	RIVER GANGA AT BEHRAMPORE (GORA BAZAR) MURSHIDABAD	WEST BENGAL	17	37	4.8	11.0	7.5	8.5	207	383	0.8	8.0	0.25	0.89	8000	240000	26000	300000	-	-
2511	NABADIP ON GANGA, GHOSHPARA NEAR MONIPURGHAT NADIA	WEST BENGAL	21	34	4.0	9.9	7.8	8.6	155	379	1.1	5.9	0.09	0.92	3000	170000	4000	220000	-	-

Station Code	LOCATION	STATE	Temperature (°C)		Dissolved Oxygen (mg/ L)		pH		Conductivity (µmho/ Cm)		Bio-Chemical Oxygen Demand (mg/L)		Nitrate (mg/ L)		Faecal Coliform (MPN/ 100 mL)		Total Coliform (MPN/ 100 mL)		Faecal Streptococci (MPN/ 100 mL)	
					< 5.0 mg/L		6.5 - 8.5				< 3.0 mg/L				< 2500 MPN/ 100 mL					
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2506	TRIBENI ON GANGA, NEAR BURNING GHAT, HOOGHLY	WEST BENGAL	22	34	4.6	9.7	7.7	8.4	217	359	1.2	6.1	0.17	0.77	2300	220000	4000	809000	-	-
10161	RIVER GANGA AT PALTA, SHITALTALA,	WEST BENGAL	21	33	3.5	9.0	7.4	8.6	215	384	1.1	6.8	0.15	0.72	39000	1300000	50000	3900000	-	-
1004	GANGA AT PALTA	WEST BENGAL	21	33	4.2	9.4	7.4	8.5	206	384	1.1	6.8	0.09	1.05	11000	220000	10000	900000	-	-
1472	GANGA AT SERAMPORE, HOOGHLY,WEST BENGAL	WEST BENGAL	21	32	4.3	11.5	7.0	8.8	214	372	1.1	6.1	0.23	0.85	4060	240000	7300	500000	-	-
1053	GANGA AT DAKSHINESHWAR, KOLKATA, WEST BENGAL	WEST BENGAL	20	34	4.7	7.8	7.1	8.6	252	569	2.7	5.5	0.14	2.15	33000	900000	49000	1600000	-	-
1471	GANGA AT HOWRAH SHIVPUR WEST BENGAL	WEST BENGAL	19	35	3.7	7.7	7.3	7.9	262	575	1.7	6.4	0.05	3.01	26000	240000	80000	500000	-	-
1470	GANGA AT GARDEN REACH, KOLKATA, WEST BENGAL	WEST BENGAL	18	34	3.8	7.0	7.3	9.0	252	522	1.2	5.3	0.26	2.48	23000	900000	70000	1600000	-	-
1052	GANGA AT ULUBERIA , HOWRAH,WEST BENGAL	WEST BENGAL	21	35	4.0	7.5	7.2	8.0	301	626	1.4	5.4	0.05	6.32	2100	110000	11000	230000	-	-
1469	GANGA AT DIAMOND HARBOUR, 24 PARGANAS (S)WEST BENGAL	WEST BENGAL	19	36	4.3	7.8	6.8	8.0	319	8049	1.2	4.1	0.09	3.81	400	11000	1100	220000	-	-
1335	GANGA AT PATHKALI NEAR DURGA CHAK WEST BENGAL	WEST BENGAL	19	35	3.7	8.3	6.8	8.5	360	18000	0.5	3.9	0.07	3.67	2300	33500	11000	90000	-	-

CHAPTER-4:
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 METHODOLOGY AND GOAL

WQI relies on normalization, the data parameter-by-parameter, as per the predicted concentration levels, and the interpretation of “bad” versus “good” levels. After this, the index is calculated as a weighted average for all observed values, with weighted parameters according to their perceived significance to overall water quality. The purposes of the WQI method are, particularly, for the evaluation of the overall status of water quality (parameters of physical, biological, and chemical) and the use of water resources for multiple purposes.

We will see the purity, and turbidity of water and can see if it’s usable or not, if not how can we make it usable. The selection of significant water quality parameters is vital and key to having a good representation of all indicators of water quality. Water quality parameters commonly used by various researchers include dissolved oxygen, total phosphates, temperature, pH, turbidity, chemical oxygen demand, fecal coliform, total solids, biochemical oxygen demand, and nitrates. The weight associated with each parameter is based on its respective standards and the magnitude of the assigned weight indicates the parameter’s significance and impact on the index.

In this application we have taken the 2 datasets and saved them in the application then we have given the option for the predefined location or manual entry where we compare the 2 to give the WQI accordingly with an option visual representation of data as well. We can see whether the water can be useful or not, and if it’s usable then we will check and give the possible use case as well unusable scenarios.

We can also see where we can use the water according to its quality like if it’s suitable for drinking, agriculture purposes, etc. This information would help the people living around the bank of river Ganga to be safe and yield high output.

4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS

The modules are designed in the following ways:

1. **Languageselection.java**

   This screen is where we select the language of the application, we have 2 Options at the moment that are English & Hindi. It is also the Primary Home screen of our application and its skippable if previous data is available.

2. **Locationselection.java**

   This screen is where we select the predefined 95 locations in the drop-down menu to get the data of water of the application, we have search functionality in the drop-down menu and it works for both the languages. It is the Secondary Home screen of our application and automatically starts if previous data is available.

3. **Enterwaterinfo.java**

   This screen is where we enter the water details manually and it works only when all the available values are filled to give the water, quality details.

4. **InputFilterMinMax.java**

   It is the filter for minimum and maximum value in the Enterwaterinfo.java here in the Edit text if you enter a data which is out of bound of possible value it restricts the user from doing so.

5. **GraphActivity.java**

   This screen is where we have the graphical view of the available 95 entries where we have 8 different graphs for each parameter that are available to us, it also has zoom and scroll functionality.

6. Res Folder

It contains all the resources like images, strings (where Hindi and English data is present), values, etc.

4.3 SOFTWARE ARCHITECTURAL DESIGNS

In this application we have taken the 2 datasets (one having the locations and its parameter and the second has all the parameters to verify WQI based on the available parameters) and saved them in the application then we have given the option for the predefined location or manual entry where we compare the 2 to give the WQI accordingly with an option visual representation of data as well. We will see the purity, and turbidity of water and can see if it is usable or not.

1. First, our main goal was to find the correct database to use for the following WQI comparison and the actual data of water bodies.
2. Then we created a blank application and started with the home screen which is language selection and added code to check if the application has any previous data or not if yes then skip this screen else wait for user input. We also added support for light and dark modes in the app.
3. Then we had to add the drop-down menu for the 95 available locations with search functionality and make it look like a home screen i.e. make it clean and user friendly.
4. After this we linked this secondary home screen to 3 options which are for going back to language screen to change the language of the app, the manual entry screen where we have data to be entered by the user manually for WQI results and lastly is the Graph screen which shows the change in parameter value along the 95 predefined location.
5. After the flow of control was set, we added the dual language functionality.
6. We then added the code to compare the WQI database which defines if the water is usable or not and for which purpose to compare to the data entered by the user in the manual as well as 95 predefined location and to return the result for it.

7. Lastly, we did all the Exception handling, Testing and debugging along with Integrated Unit Testing as well which then made our application fully stable to use.

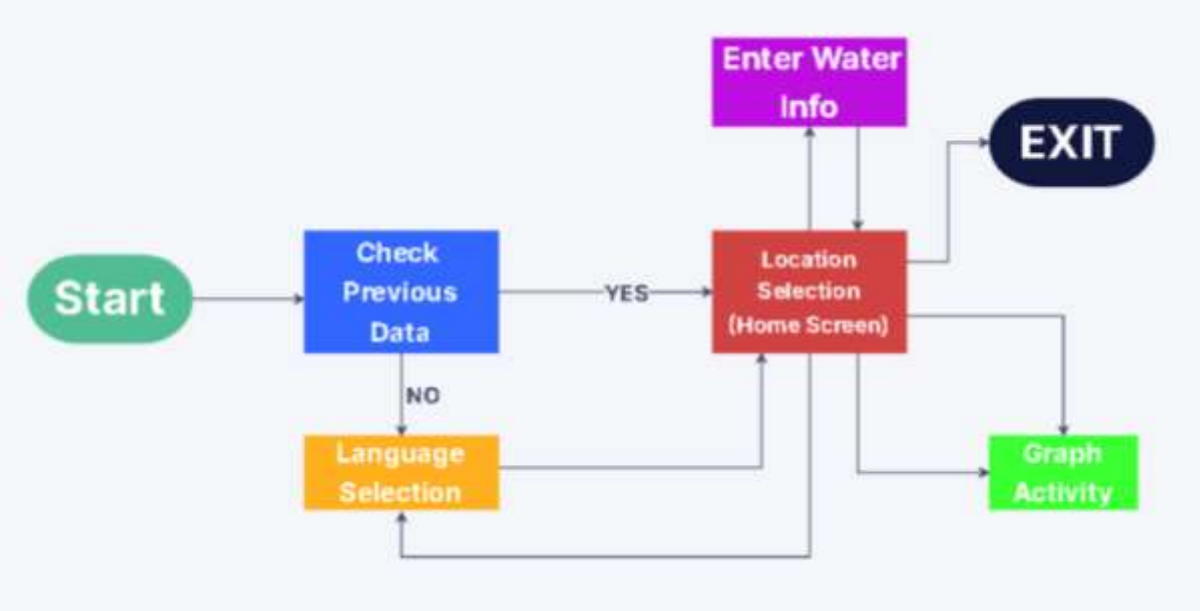

Figure 4.3.1 System Architecture

4.4 USER INTERFACE DESIGNS

While developing this application our main goal was to make an application that is so user-friendly that any individual can make the application run and work without giving any set of instructions to them. We wanted it to be interactive and smart hence we have added the graphical view along with support for light and dark mode.

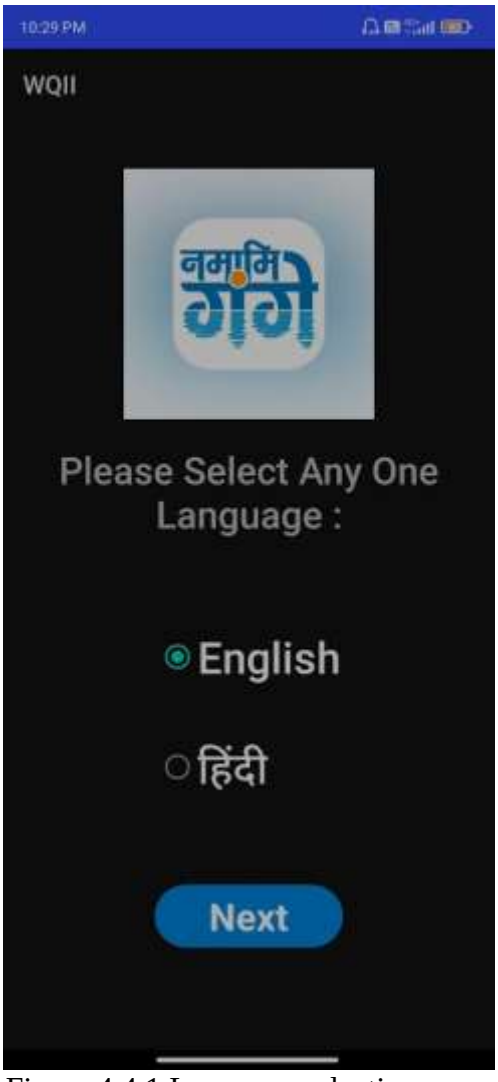

Figure 4.4.1 Language selection screen

Figure 4.4.2 Location selection screen

Figure 4.4.3 Enter water info screen

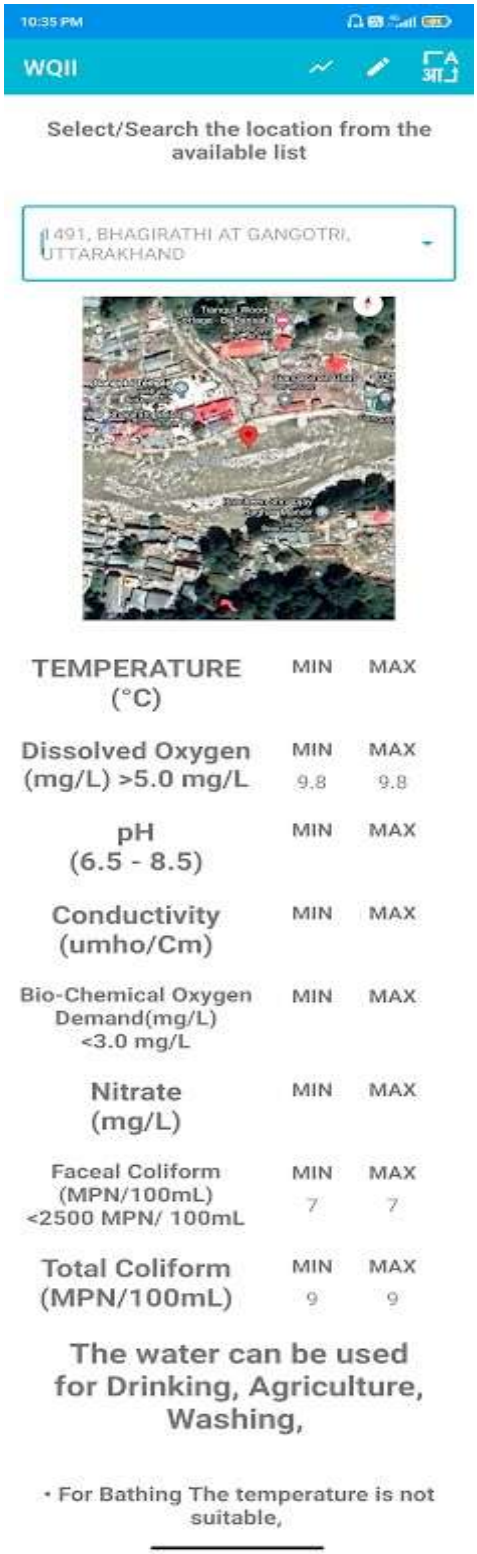

Figure 4.4.4 Location selection screen

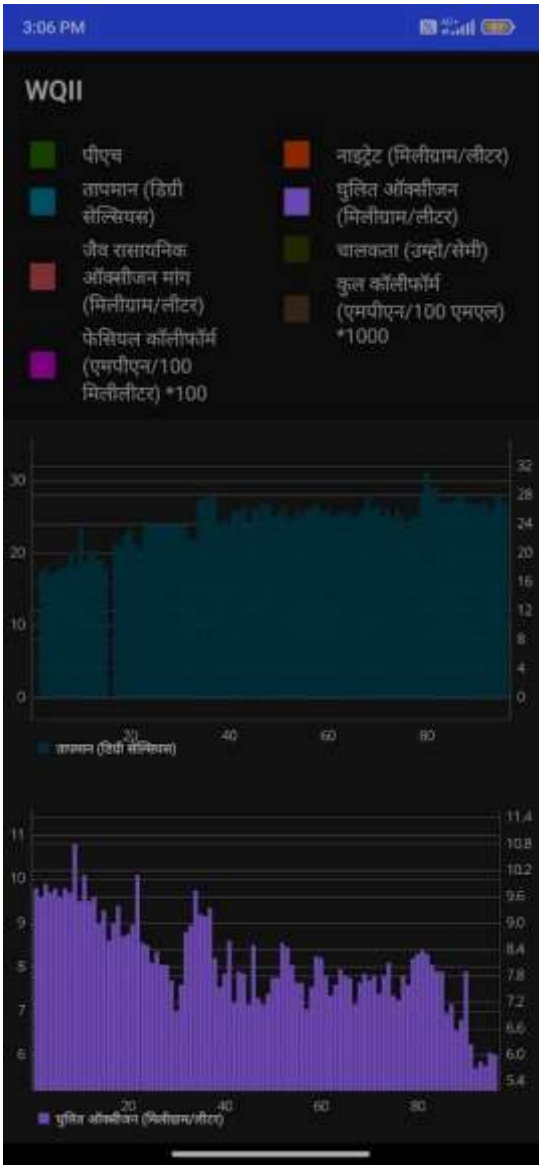

Figure 4.4.5 Graph activity dark mode

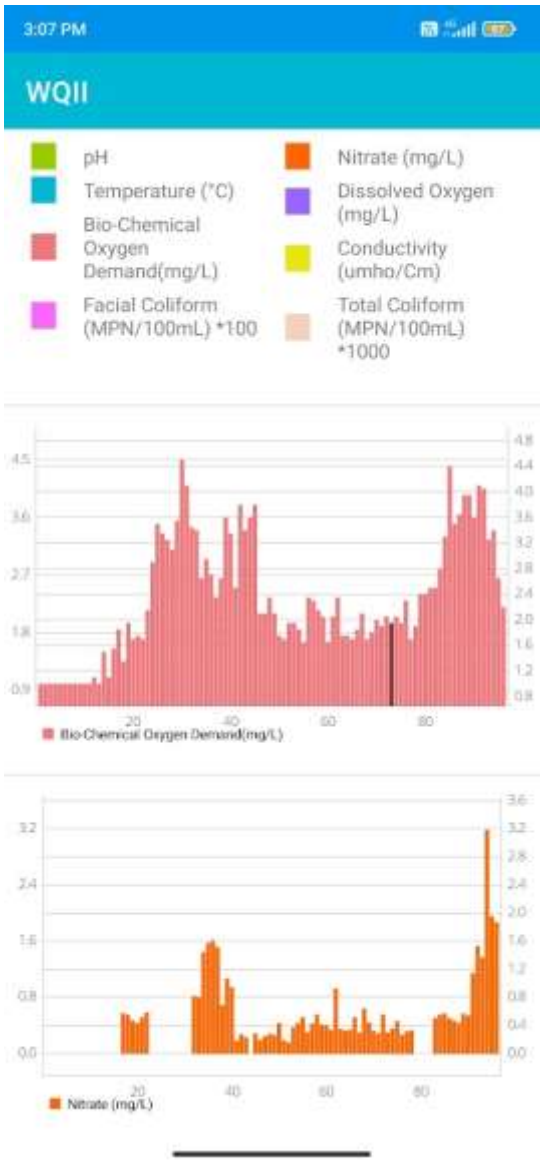

Figure 4.4.6 Graph activity light mode

CHAPTER – 5

5.1 TECHNICAL CODING AND CODE SOLUTIONS

We have made the application successfully and the code along with the details are available on GitHub we have given the link for the code below and it's ready to be live but we haven't done that at the moment.

- **Languageselection.java**

This screen is where we select the language of the application, we have 2 Options at the moment that are English & Hindi. It is also the Primary Home screen of our application and its skippable if previous data is available.

```
package io.github.pv.wqii_final;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.RadioButton;
import android.widget.TextView;

public class Languageselection extends AppCompatActivity {

   private RadioButton radioenglishButton;
   private RadioButton radiohindiButton;

   @Override
```

```
  protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_languageselection);
     CardView btnDisplay = (CardView) findViewById(R.id.next);
     SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref",
MODE_PRIVATE);
     SharedPreferences.Editor editor = pref.edit();
     if(pref.getInt("Previousdata",0)!=0)
     {
        Intent intent1 = new Intent(getApplicationContext(), Locationselection.class);
        startActivity(intent1);
        finish();

     }
     radioenglishButton =(RadioButton)findViewById(R.id.en);
     if(pref.getString("Language", "").equals("Hindi"))
     {
        TextView languageask=(TextView) findViewById(R.id.languagequestion);
        languageask.setText("कृपया कोई एक भाषा चुनें:");
        TextView nexttext=(TextView) findViewById(R.id.nexttext);
        nexttext.setText("अगला");


        radiohindiButton=(RadioButton)findViewById(R.id.hin);
        radioenglishButton.setChecked(false);
        radiohindiButton.setChecked(true);

     }
```

```
    btnDisplay.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            if(radioenglishButton.isChecked())
            {
                editor.putInt("Previousdata", 1);
                editor.putString("Language", "English");

            }
            else
            {
                editor.putInt("Previousdata", 1);
                editor.putString("Language", "Hindi");

            }

            editor.putInt("Firststart", 1);
            editor.apply();
            Intent intent1 = new Intent(getApplicationContext(), Locationselection.class);
            startActivity(intent1);
            finish();

        }
    });

}
@Override
public void onBackPressed() {

    SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref",
```

```
MODE_PRIVATE);
        SharedPreferences.Editor editor = pref.edit();
        if(pref.getInt("Firststart",0)==0) {
            finish();
        }

        else {
            editor.putInt("Previousdata", 1);
            editor.apply();
            Intent intent1 = new Intent(getApplicationContext(), Locationselection.class);
            startActivity(intent1);
            finish();
        }

    }
}
```

- **InputFilterMinMax.java –**

It is the filter for minimum and maximum value in the Enterwaterinfo.java here in the Edit text if you enter a data which is out of bound of possible value it restricts the user from doing so.

```
package io.github.pv.wqii_final;

import android.text.InputFilter;
    import android.text.Spanned;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class InputFilterMinMax implements InputFilter {
```

```
private float min, max;

public InputFilterMinMax(float min, float max) {
   this.min = min;
   this.max = max;
}

public InputFilterMinMax(String min, String max) {
   this.min = Float.parseFloat(min);
   this.max = Float.parseFloat(max);
}

@Override
public CharSequence filter(CharSequence source, int start, int end, Spanned dest, int dstart,
 int dend) {
   try {
      // Remove the string out of destination that is to be replaced
      String newVal = dest.toString().substring(0, dstart) + dest.toString().substring(dend,
 dest.toString().length());
      // Add the new string in
      newVal = newVal.substring(0, dstart) + source.toString() + newVal.substring(dstart,
 newVal.length());
      float input = Float.parseFloat(newVal);
      Pattern mPattern = Pattern.compile("[0-9]{0," + (10 - 1) + "}+((\\.[0-9]{0," + (2 - 1) +
 "})?)||(\\.)?");

      Matcher matcher = mPattern.matcher(dest);
      if (isInRange(min, max, input) && matcher.matches())
         return null;
   } catch (NumberFormatException nfe) { }
   return "";
```

```
}
private boolean isInRange(float a, float b, float c) {
    return b > a ? c >= a && c <= b : c >= b && c <= a;
}
  }
```

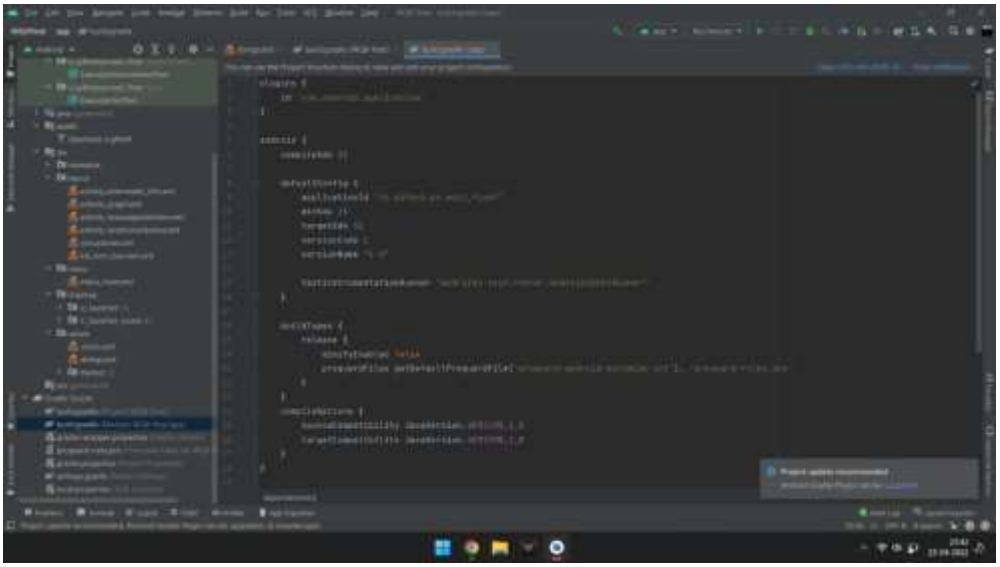

Figure 5.1.1 Build Gradle

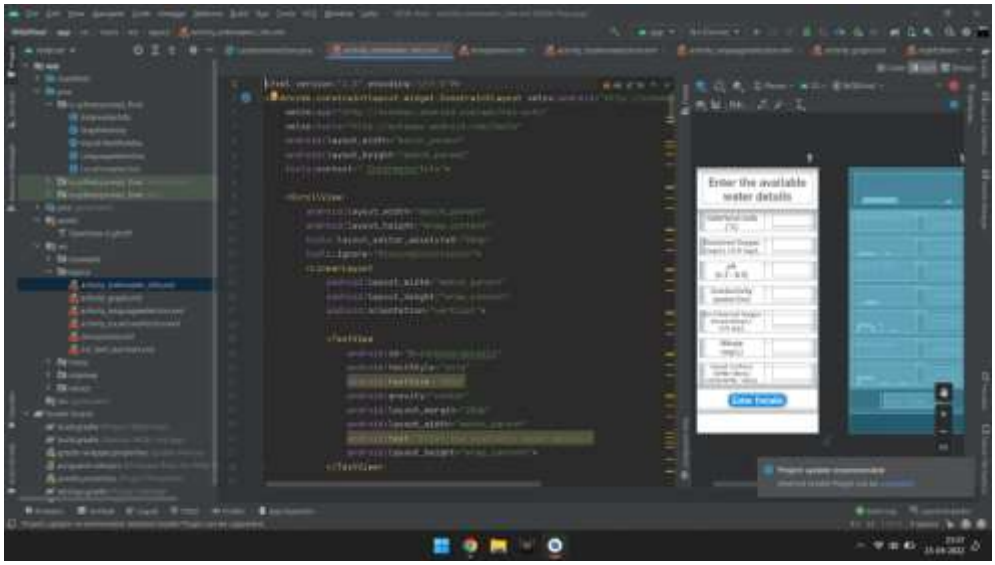

Figure 5.1.2 Enterwaterinfo.java

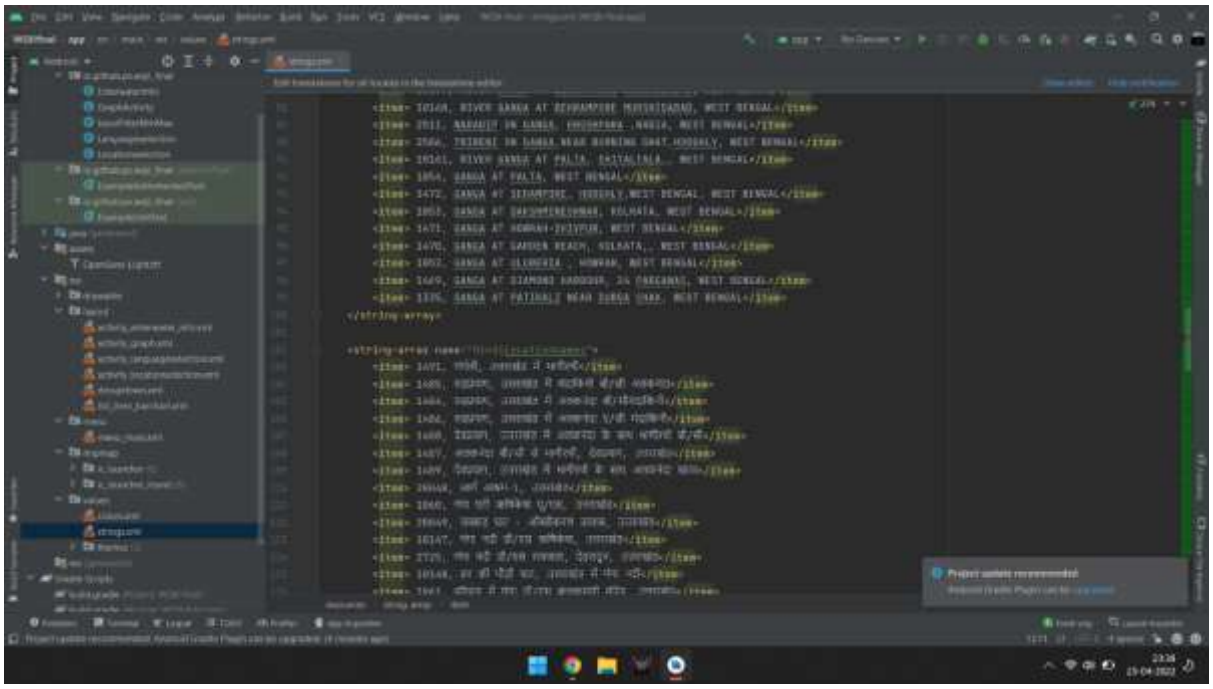

Figure 5.1.3 Hindi/English string array

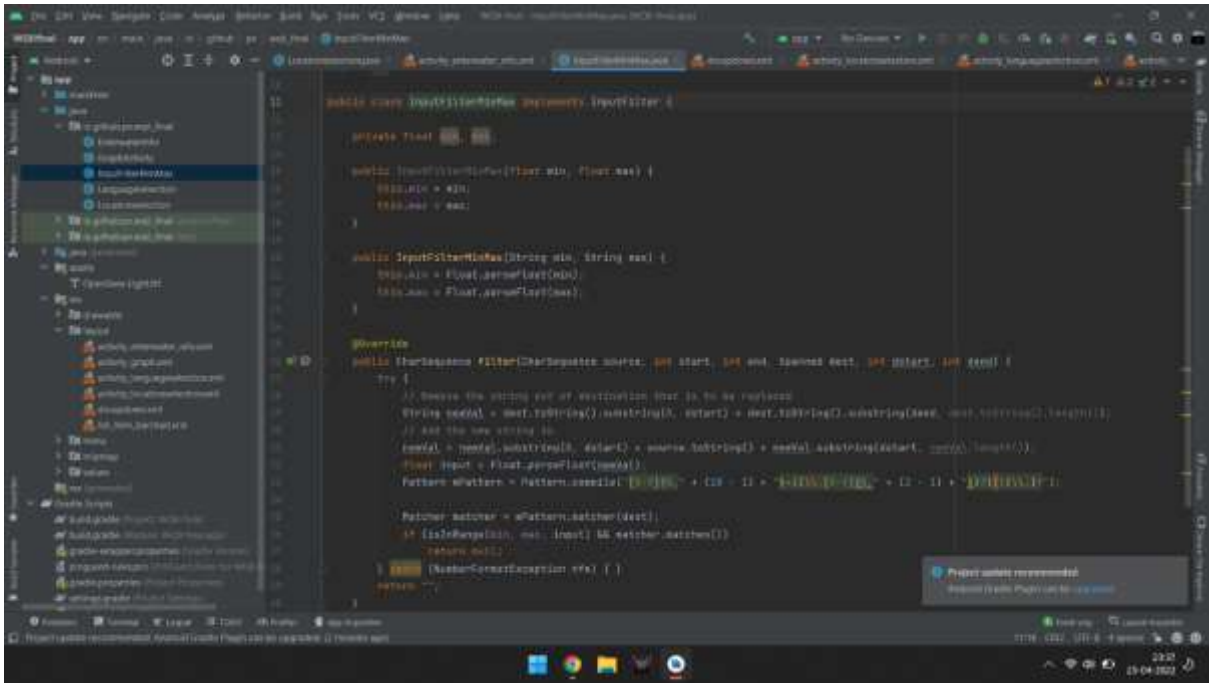

Figure 5.1.4 FilterMinMax.java

5.2 PROTOTYPE SUBMISSION

We have made the application successfully and it's ready to be live we haven't done that at the moment but we have given the link to the release version of the app to be downloaded and installed on Android devices.

Link to the apk –WQII.apk
Demo video link – Video link
Project link – GitHub link

5.3 TEST AND VALIDATION

We have done exception handling (like the min max function which limits the user from entering wrong values in the manual mode) in the application itself along with that for the android application we have something called Unit & UI Testing.

- **Integrated Unit Testing** – We have done them using the Integrated Unit testing via the inbuilt **Espresso test** that the android studio provides which helps us to know that the application is stable to use.

```
package io.github.pv.wqii_final;


import static androidx.test.espresso.Espresso.onData;

import static androidx.test.espresso.Espresso.onView;

import static androidx.test.espresso.Espresso.pressBack;

import static androidx.test.espresso.action.ViewActions.click;

import static androidx.test.espresso.action.ViewActions.closeSoftKeyboard;

import static androidx.test.espresso.action.ViewActions.pressImeActionButton;

import static androidx.test.espresso.action.ViewActions.replaceText;

import static androidx.test.espresso.action.ViewActions.scrollTo;
```

```
import static androidx.test.espresso.matcher.ViewMatchers.isDisplayed;
import static androidx.test.espresso.matcher.ViewMatchers.withClassName;
import static androidx.test.espresso.matcher.ViewMatchers.withContentDescription;
import static androidx.test.espresso.matcher.ViewMatchers.withId;
import static androidx.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;
import static org.hamcrest.Matchers.anything;
import static org.hamcrest.Matchers.is;

import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import androidx.test.espresso.DataInteraction;
import androidx.test.espresso.ViewInteraction;
import androidx.test.filters.LargeTest;
import androidx.test.rule.ActivityTestRule;
import androidx.test.runner.AndroidJUnit4;

import org.hamcrest.Description;
import org.hamcrest.Matcher;
import org.hamcrest.TypeSafeMatcher;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;
```

```
@LargeTest
@RunWith(AndroidJUnit4.class)
public class LanguageselectionTest {

   @Rule
   public ActivityTestRule<Languageselection> mActivityTestRule = new
ActivityTestRule<>(Languageselection.class);

   @Test
   public void languageselectionTest() {
      ViewInteraction cardView = onView(
            allOf(withId(R.id.next),
                  childAtPosition(
                        childAtPosition(
                              withId(android.R.id.content),
                              0),
                        7),
                  isDisplayed()));
      cardView.perform(click());

      ViewInteraction actionMenuItemView = onView(
            allOf(withId(R.id.graphdata), withContentDescription("GraphInput"),
                  childAtPosition(
                        childAtPosition(
```

```
                withId(androidx.appcompat.R.id.action_bar),
                1),
            0),
        isDisplayed()));
    actionMenuItemView.perform(click());

    pressBack();

    ViewInteraction materialAutoCompleteTextView = onView(
        allOf(withId(R.id.selectlocation),
            childAtPosition(
                childAtPosition(
withClassName(is("com.google.android.material.textfield.TextInputLayout")),
                    0),
                0)));
    materialAutoCompleteTextView.perform(scrollTo(), click());

    DataInteraction materialTextView = onData(anything())
        .inAdapterView(childAtPosition(
            withClassName(is("android.widget.PopupWindow$PopupBackgroundView")),
            0))
        .atPosition(8);
    materialTextView.perform(click());
```

```
        ViewInteraction materialAutoCompleteTextView2 = onView(
                allOf(withId(R.id.selectlocation),
                        childAtPosition(
                                childAtPosition(
withClassName(is("com.google.android.material.textfield.TextInputLayout")),
                                        0),
                                0)));
        materialAutoCompleteTextView2.perform(scrollTo(), click());

        ViewInteraction materialAutoCompleteTextView3 = onView(
                allOf(withId(R.id.selectlocation),
                        childAtPosition(
                                childAtPosition(
withClassName(is("com.google.android.material.textfield.TextInputLayout")),
                                        0),
                                0)));
        materialAutoCompleteTextView3.perform(scrollTo(), replaceText("1"),
closeSoftKeyboard());

        DataInteraction materialTextView2 = onData(anything())
                .inAdapterView(childAtPosition(
                        withClassName(is("android.widget.PopupWindow$PopupBackgroundView")),
                        0))
                .atPosition(5);
```

```
materialTextView2.perform(click());

ViewInteraction actionMenuItemView2 = onView(
        allOf(withId(R.id.addwaterdata), withContentDescription("EditUserInput"),
                childAtPosition(
                        childAtPosition(
                                withId(androidx.appcompat.R.id.action_bar),
                                1),
                        1),
                isDisplayed()));
actionMenuItemView2.perform(click());

ViewInteraction appCompatEditText = onView(
        allOf(withId(R.id.entertemp),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                0),
                        2)));
appCompatEditText.perform(scrollTo(), replaceText("1"), closeSoftKeyboard());

ViewInteraction appCompatEditText2 = onView(
        allOf(withId(R.id.entertemp), withText("1"),
                childAtPosition(
                        childAtPosition(
```

```
                withClassName(is("android.widget.LinearLayout")),
                0),
            2)));
appCompatEditText2.perform(pressImeActionButton());

ViewInteraction appCompatEditText3 = onView(
    allOf(withId(R.id.enterdissolvedoxygen),
        childAtPosition(
            childAtPosition(
                withClassName(is("android.widget.LinearLayout")),
                1),
            2)));
appCompatEditText3.perform(scrollTo(), replaceText("8"), closeSoftKeyboard());

ViewInteraction appCompatEditText4 = onView(
    allOf(withId(R.id.enterdissolvedoxygen), withText("8"),
        childAtPosition(
            childAtPosition(
                withClassName(is("android.widget.LinearLayout")),
                1),
            2)));
appCompatEditText4.perform(pressImeActionButton());

ViewInteraction appCompatEditText5 = onView(
    allOf(withId(R.id.enterpH),
```

```
            childAtPosition(
                childAtPosition(
                    withClassName(is("android.widget.LinearLayout")),
                    2),
                2)));
appCompatEditText5.perform(scrollTo(), replaceText("4"), closeSoftKeyboard());

ViewInteraction appCompatEditText6 = onView(
    allOf(withId(R.id.enterpH), withText("4"),
            childAtPosition(
                childAtPosition(
                    withClassName(is("android.widget.LinearLayout")),
                    2),
                2)));
appCompatEditText6.perform(pressImeActionButton());

ViewInteraction appCompatEditText7 = onView(
    allOf(withId(R.id.enterConductivity),
            childAtPosition(
                childAtPosition(
                    withClassName(is("android.widget.LinearLayout")),
                    3),
                2)));
appCompatEditText7.perform(scrollTo(), replaceText("5"), closeSoftKeyboard());
```

```
ViewInteraction appCompatEditText8 = onView(
        allOf(withId(R.id.enterConductivity), withText("5"),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                3),
                        2)));
appCompatEditText8.perform(pressImeActionButton());

ViewInteraction appCompatEditText9 = onView(
        allOf(withId(R.id.enterbiochemoxygen),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                4),
                        2)));
appCompatEditText9.perform(scrollTo(), replaceText("4"), closeSoftKeyboard());

ViewInteraction appCompatEditText10 = onView(
        allOf(withId(R.id.enterbiochemoxygen), withText("4"),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                4),
                        2)));
```

```
appCompatEditText10.perform(pressImeActionButton());

ViewInteraction appCompatEditText11 = onView(
        allOf(withId(R.id.enterNitrate),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                5),
                        2)));
appCompatEditText11.perform(scrollTo(), replaceText("5"), closeSoftKeyboard());

ViewInteraction appCompatEditText12 = onView(
        allOf(withId(R.id.enterNitrate), withText("5"),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                5),
                        2)));
appCompatEditText12.perform(pressImeActionButton());

ViewInteraction appCompatEditText13 = onView(
        allOf(withId(R.id.enterFacealColiform),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
```

```
                6),
            2)));
appCompatEditText13.perform(scrollTo(), replaceText("5"), closeSoftKeyboard());

ViewInteraction appCompatEditText14 = onView(
    allOf(withId(R.id.enterFacealColiform), withText("5"),
        childAtPosition(
            childAtPosition(
                withClassName(is("android.widget.LinearLayout")),
                6),
            2)));
appCompatEditText14.perform(pressImeActionButton());

ViewInteraction cardView2 = onView(
    allOf(withId(R.id.next),
        childAtPosition(
            childAtPosition(
                withClassName(is("android.widget.ScrollView")),
                0),
            3)));
cardView2.perform(scrollTo(), click());

ViewInteraction appCompatEditText15 = onView(
    allOf(withId(R.id.enterFacealColiform), withText("5"),
        childAtPosition(
```

```
                childAtPosition(
                        withClassName(is("android.widget.LinearLayout")),
                        6),
                2)));
appCompatEditText15.perform(scrollTo(), click());

ViewInteraction appCompatEditText16 = onView(
        allOf(withId(R.id.enterFacealColiform), withText("5"),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                6),
                        2)));
appCompatEditText16.perform(scrollTo(), click());

ViewInteraction appCompatEditText17 = onView(
        allOf(withId(R.id.enterFacealColiform), withText("5"),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                6),
                        2)));
appCompatEditText17.perform(scrollTo(), replaceText(""));

ViewInteraction appCompatEditText18 = onView(
```

```
        allOf(withId(R.id.enterFacealColiform),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                6),
                        2),
                isDisplayed()));
appCompatEditText18.perform(closeSoftKeyboard());

ViewInteraction appCompatEditText19 = onView(
        allOf(withId(R.id.enterFacealColiform),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.LinearLayout")),
                                6),
                        2)));
appCompatEditText19.perform(pressImeActionButton());

ViewInteraction cardView3 = onView(
        allOf(withId(R.id.next),
                childAtPosition(
                        childAtPosition(
                                withClassName(is("android.widget.ScrollView")),
                                0),
                        3)));
```

```
cardView3.perform(scrollTo(), click());

pressBack();

ViewInteraction actionMenuItemView3 = onView(
        allOf(withId(R.id.language), withContentDescription("EditUserInput"),
                childAtPosition(
                        childAtPosition(
                                withId(androidx.appcompat.R.id.action_bar),
                                1),
                        2),
                isDisplayed()));
actionMenuItemView3.perform(click());

ViewInteraction materialRadioButton = onView(
        allOf(withId(R.id.hin), withText("?????"),
                childAtPosition(
                        allOf(withId(R.id.radioGrp),
                                childAtPosition(
                                        withClassName(is("android.widget.LinearLayout")),
                                        0)),
                        2),
                isDisplayed()));
materialRadioButton.perform(click());
```

```
        ViewInteraction cardView4 = onView(
                allOf(withId(R.id.next),
                        childAtPosition(
                                childAtPosition(
                                        withId(android.R.id.content),
                                        0),
                                7),
                        isDisplayed()));
        cardView4.perform(click());

        ViewInteraction materialAutoCompleteTextView4 = onView(
                allOf(withId(R.id.selectlocation),
                        childAtPosition(
                                childAtPosition(
withClassName(is("com.google.android.material.textfield.TextInputLayout")),
                                        0),
                                0)));
        materialAutoCompleteTextView4.perform(scrollTo(), click());

        DataInteraction materialTextView3 = onData(anything())
                .inAdapterView(childAtPosition(
                        withClassName(is("android.widget.PopupWindow$PopupBackgroundView")),
                        0))
                .atPosition(21);
```

```
        materialTextView3.perform(click());
    }

    private static Matcher<View> childAtPosition(
            final Matcher<View> parentMatcher, final int position) {

        return new TypeSafeMatcher<View>() {
            @Override
            public void describeTo(Description description) {
                description.appendText("Child at position " + position + " in parent ");
                parentMatcher.describeTo(description);
            }

            @Override
            public boolean matchesSafely(View view) {
                ViewParent parent = view.getParent();
                return parent instanceof ViewGroup && parentMatcher.matches(parent)
                        && view.equals(((ViewGroup) parent).getChildAt(position));
            }
        };
    }
}
```

CHAPTER-6:

PROJECT OUTCOME AND APPLICABILITY

6.1 OUTLINE

We have successfully created an app that is suitable for English as well as Hindi user along with that we have 95 predefined locations where Ganga flows and has search functionality and have also included the manual option for entering the water quality for any water body, and lastly, we have added a graphical view for more interactive to the user.

6.2 PROJECT APPLICABILITY TO REAL-WORLD APPLICATIONS

1. We have the functional app with all the work that we wanted to provide for most of the part we have also added graphical data for the visual representation of data as well. Improvement can be made at any time that's what we believe and would just need optimization and increasing the efficiency (130 Mb release) that would happen along with time to make it good to go live on the play store if we would like to do that or share the application to the users.

2. We can see whether the water we are using is pure or not if not, how can we make it pure or how can we use it for other purposes.

3. We will take the process to another level and also check many other parameters of water and we have successfully made the app as well supporting two languages and 2 themes with graphical data representation as well.

4. Identifying the source (s) of contamination and developing appropriate management strategies to minimize potential public health risks.

5. Data obtained via assessment and monitoring water quality provides empirical evidence to assist health and environmental decision making. In water management practices, water quality values serve as useful and sensitive indicators of changes in the physical, chemical or biological composition of the overall water status.

CHAPTER-7

CONCLUSIONS AND RECOMMENDATION

7.1 LIMITATION

Despite the benefits attributed to the WQI, it is however besieged with some challenges, some of which are stated below,

1. WQI is not an absolute measure of the degree of pollution or the actual water quality.
2. Lack of precision and accuracy in classification technique of importance of evaluation of parameters and lack of live data.
3. Inefficiency in dealing with uncertainty and subjectivity in a complex environmental issue such as the incompatibility of observations, uncertainty, and imprecision in criteria.
4. Lack of a uniform method for measuring water pollution involving biological parameters.
5. Inadequate the transfer of complex environmental data into information.

7.2 FUTURE ENHANCEMENTS

There's always going to be room for improvements and some which we think can happen or should be done.

1. Having it linked to Live Data which would help in making this procedure real-time as well as would increase the correctness and accuracy of the application.
2. Adding more available locations and covering more water bodies.
3. Decreasing the size of the application and making further optimizations.
4. More local language support.
5. Regular updates to meet the criteria for newer
6. android compatibility.

7.3 CONCLUSION

The water quality index (WQI) is the best tool for giving the details of the overall grade of water, which is a process to reduce large numbers and parameters into a single index number. WQI is very effective for understanding water quality findings and is used to judge the appropriateness of water for drinking purposes in major regions in the world. WQI is defined as a rating that reflects the composite influence of different water quality parameters. Integrating this feature in an application (WQII) with multiple languages, simplistic & user-friendly UI is going to help a lot of people to decide using the water for safe uses only which would give greater yield overall with less medical complications to the people around the water bodies.

REFERENCES

1. http://125.19.52.218/realtime/
2. https://nmcg.nic.in/NamamiGanga.aspx
3. https://developer.android.com/studio
4. https://www.sciencedirect.com/science/article/pii/S1470160X203_11572#b9010
5. https://drive.google.com/file/d/18z4n39bch_BJig7d3sqxuBfY80KISo8I/view?usp=sharing
6. https://drive.google.com/file/d/18f7rNqQ-rXsMlCDx2rcdy7TZd-b2BGPR/view