

PHASE 4: DEVELOPMENT PART – 2

PROJECT TITLE	9238 – FLOOD MONITORING AND EARLY WARNING
NAME	R.Priyadharsni
TEAM ID	5246
TEAM NAME	PROJ_204183_TEAM_1
COLLEGE CODE - NAME	9238 – MANGAYARKARASI COLLEGE OF ENGINEERING PARAVAI MADURAI.
GROUP	5
GITHUB REPOSITORY LINK	https://github.com/priyadharsni/IBM-Naanmudhalvan-iot.git

Project description:

Here initially the data from the sensors are Collected by the micro controller then it transmits the data to the Cloud through the LoRa Gateway which help to transmit the data to long range. Later the data are collected by the Cloud and Pre-Processed. it gets the real time update from the sensors and then it sends the efficient information to the user through the SMS and social media the user can see the flood area from the map. If the water level is above the threshold, then it sends the alert messages to the user.

Android studio:

Android Studio is the official integrated development environment (IDE) for Android app development. It's a powerful and feature-rich IDE developed by Google, designed to make it easier for developers to create, test, and deploy Android applications. Here, I'll provide an overview of Android Studio in detail: In Android Studio, you can create various types of files for your Android app project, including Java/Kotlin source files, XML layout files, resource files, and more. Here are the general steps to create a file in Android Studio

How to create file in android studio:

Open or Create a Project:

If you already have an Android Studio project open, make sure it's loaded in the IDE. If you're starting a new project, you can create one by going to "File" > "New" > "New Project."

Select the Destination Directory:

In the Project Explorer on the left side of the IDE, navigate to the directory where you want to create the new file. Typically, Java/Kotlin source files are placed in the "java" directory, XML layout files in the "res" directory, and other resources in their respective directories.

. Create a New File:

Right-click on the directory where you want to create the file. Select "New" to open the submenu. Choose the type of file you want to create. For example, to create a new Java or Kotlin class, select "Java Class" or "Kotlin Class."

Specify File Details:

In the dialog that appears, you'll be prompted to provide details about the file you're creating. This may include the file name, the package name for source files, and the type of file (Java class, XML layout, etc.). Fill in the necessary information and click "OK" or "Finish" to create the file.

Edit the File:

After creating the file, Android Studio will open it in the code editor. You can start editing the file by adding your code, layout components, or other resources as needed.

Save the File:

Make sure to save the file after making changes by either clicking the "Save" icon (a floppy disk icon) in the toolbar or pressing Ctrl + S (or Command + S on macOS) on your keyboard

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/back"

    android:padding="16dp"

    tools:ignore="ExtraText">

    <EditText

        android:id="@+id/emailEditText"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:hint="Email"

        android:inputType="textEmailAddress"
```

```
android:textColor="#000000"
android:textColorHint="#070707"
app:layout_constraintBottom_toTopOf="@+id/passwordEditText"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.812"
tools:layout_editor_absoluteX="16dp" />
```

<EditText

```
android:id="@+id/passwordEditText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="476dp"
android:hint="Password"
android:inputType="textPassword"
android:textColor="#0B0A0A"
android:textColorHighlight="#F8F5F5"
android:textColorHint="#000000"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.125"
app:layout_constraintStart_toStartOf="parent" />
```

<Button

```
android:id="@+id/loginButton"
android:layout_width="95dp"
android:layout_height="49dp"
android:layout_marginEnd="64dp"
android:backgroundTint="#04CEE8"
android:text="Login"
android:textColorHighlight="#CD7C7C"
android:textColorLink="#0ADB6F"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.419"
app:rippleColor="#E14545"
app:strokeColor="#E85656" />
```

```
<Button
    android:id="@+id/signup"
    android:layout_width="101dp"
    android:layout_height="46dp"
    android:layout_marginEnd="24dp"
    android:backgroundTint="#09CCE4"
    android:text="signup"
    android:textColorHighlight="#DC4040"
    android:textColorLink="#D84577"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/loginButton"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.42" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="251dp"
    android:layout_height="36dp"
    android:layout_marginBottom="24dp"
    android:fontFamily="sans-serif-black"
    android:text="IoT Flood Login"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textColorHighlight="#FAF9F9"
```

```
android:textColorHint="#FFFFFF"
android:textColorLink="#FBF9F9"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBottom_toTopOf="@+id/emailEditText"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.506"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.859" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```



Fig: main page

Activity_login.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/flood"
    android:padding="16dp">

    <EditText
android:id="@+id/emailEditText"

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:textColor="#0E0E0E"
        app:layout_constraintBottom_toTopOf="@+id/passwordEditText"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.844"
        tools:layout_editor_absoluteX="16dp" />

    <EditText

        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="436dp"
        android:hint="Password"
        android:inputType="textPassword"
```

```
app:layout_constraintBottom_toBottomOf="parent"
tools:layout_editor_absoluteX="16dp" />
```

```
<Button
```

```
    android:id="@+id/registerButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="#97BA99"
    android:text="Register"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.803"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/passwordEditText"
    app:layout_constraintVertical_bias="0.094" />
```

```
<TextView
```

```
    android:id="@+id/textView2"
    android:layout_width="286dp"
    android:layout_height="69dp"
    android:text="IOT flood signup"
    android:textAlignment="center"
    android:textAllCaps="true"
    android:textColor="#FFBFBF"
    android:textColorHighlight="#D9EEF1"
    android:textColorHint="#151515"
    android:textSize="24sp"
    android:textStyle="bold"
    tools:layout_editor_absoluteX="62dp"
    tools:layout_editor_absoluteY="16dp" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```




Fig: login page

Activity_dashboard.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
```



```
android:layout_height="match_parent"
android:background="@drawable/dam"
android:orientation="vertical"
android:padding="16dp"
tools:context=".DashboardActivity">
```

<Button

```
    android:id="@+id/paymentButton"
    android:layout_width="323dp"
    android:layout_height="95dp"
    android:backgroundTint="#0C0C0C"
    android:text="waterlevel"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.515" />
```

<Button

```
    android:id="@+id/settingsButton"
    android:layout_width="318dp"
    android:layout_height="110dp"
    android:layout_marginBottom="136dp"
    android:backgroundTint="#100F0F"
    android:text="Settings"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.645"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/floodrange"
    app:layout_constraintVertical_bias="0.197"
```

```
tools:ignore="UnknownId" />
```

```
<Button
```

```
    android:id="@+id/myReservationsButton"
    android:layout_width="160dp"
    android:layout_height="72dp"
    android:layout_marginTop="36dp"
    android:backgroundTint="#191717"
    android:text="weatherforecast"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.924"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3"
    app:layout_constraintVertical_bias="0.159" />
```

```
<TextView
```

```
    android:id="@+id/textView3"
    android:layout_width="385dp"
    android:layout_height="73dp"
    android:padding="16dp"
    android:text="floodmonitoring dashboard"
    android:textAlignment="center"
    android:textColor="#121010"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/paymentButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.769"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

<Button

```
    android:id="@+id/findParkingButton"
    android:layout_width="159dp"
    android:layout_height="73dp"
    android:layout_marginTop="36dp"
    android:backgroundTint="#100F0F"
    android:text="flow of water"
    android:textColorHighlight="#191717"
    android:textColorLink="#1B1717"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/myReservationsButton"
    app:layout_constraintHorizontal_bias="0.355"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3"
    app:layout_constraintVertical_bias="0.163" />
```

<ImageView

```
    android:id="@+id/imageView"
    android:layout_width="39dp"
    android:layout_height="69dp"
    android:layout_marginStart="4dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/textView3"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/icon" />
```

<TextView

```
    android:id="@+id/textView4"
```

```
android:layout_width="294dp"
android:layout_height="44dp"
android:backgroundTint="#0C0C0C"
android:textAlignment="gravity"
android:textAllCaps="true"
android:textColor="#F6F3F3"
android:textSize="12sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.957"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.181" />
```

<ImageView

```
android:id="@+id/imageView4"
android:layout_width="40dp"
android:layout_height="69dp"
android:layout_marginEnd="4dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="@+id/textView3"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0"
app:srcCompat="@drawable/icon" />
```

<ImageView

```
android:id="@+id/imageView7"
android:layout_width="60dp"
android:layout_height="69dp"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.034"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.164"
app:srcCompat="@drawable/person" />
```

<!-- Add more buttons as needed -->

</androidx.constraintlayout.widget.ConstraintLayout>



Fig:dashboard page

activity_firebase.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000C33"
    tools:context=".Firebase">

    <!-- Custom Meter Gauge -->

    <com.example.floodmonitoring.GaugeView
        android:id="@+id/gaugeView"
        android:layout_width="328dp"
        android:layout_height="254dp"
        android:layout_centerInParent="true"
        android:rotation="0"
        android:rotationY="0"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.033" />
```

```
<com.example.floodmonitoring.GaugeView
    android:id="@+id/gaugeView3"
    android:layout_width="328dp"
    android:layout_height="254dp"
    android:layout_centerInParent="true"
    android:rotation="0"
    android:rotationY="0"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.517" />
```

```
<com.example.floodmonitoring.GaugeView
    android:id="@+id/gaugeView4"
    android:layout_width="328dp"
    android:layout_height="254dp"
    android:layout_centerInParent="true"
    android:rotation="0"
    android:rotationY="0"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />
```

```
<TextView
    android:id="@+id/textView4"
```



```
    android:layout_width="243dp"
    android:layout_height="39dp"
    android:text="Water Level"
    android:textAlignment="center"
    android:textColor="#FDFDFD"
    android:textSize="32sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.267" />
```

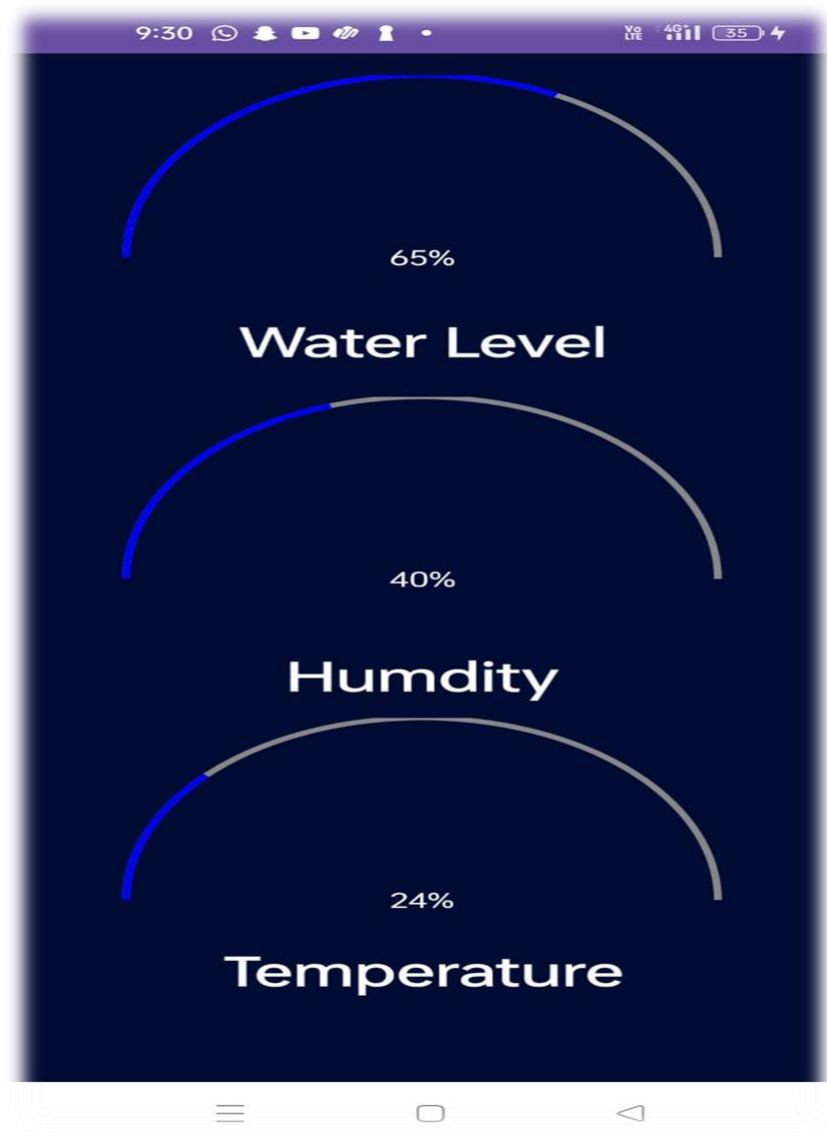
<TextView

```
    android:id="@+id/textView6"
    android:layout_width="243dp"
    android:layout_height="39dp"
    android:text="Humdity"
    android:textAlignment="center"
    android:textColor="#FDFDFD"
    android:textSize="32sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.611" />
```

<TextView

```
    android:id="@+id/textView8"
    android:layout_width="243dp"
```

```
        android:layout_height="39dp"
        android:text="Temperature"
        android:textAlignment="center"
        android:textColor="#FDFDFD"
        android:textSize="32sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.914" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



activity_weather.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#200025"
    tools:context=".Weather">

    <EditText
        android:id="@+id/locationEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:hint="Enter your location"
        android:textColor="#FFFFFF" />

    <Button
        android:id="@+id/getWeatherButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/locationEditText"
        android:layout_centerHorizontal="true"
        android:text="Get Weather" />

    <TextView
```

```
android:id="@+id/locationTextView"
android:layout_width="214dp"
android:layout_height="42dp"
android:layout_below="@id/getWeatherButton"
android:layout_alignParentStart="true"
android:layout_alignParentEnd="true"
android:layout_marginStart="98dp"
android:layout_marginTop="49dp"
android:layout_marginEnd="99dp"
android:text="Location"
android:textAlignment="center"
android:textColor="#FFFFFF"
android:textColorHint="#957373" />
```

<TextView

```
android:id="@+id/temperatureTextView"
android:layout_width="210dp"
android:layout_height="34dp"
android:layout_below="@id/locationTextView"
android:layout_alignParentStart="true"
android:layout_alignParentEnd="true"
android:layout_marginStart="100dp"
android:layout_marginTop="88dp"
android:layout_marginEnd="101dp"
android:text="Temperature"
android:textAlignment="center"
android:textColor="#FFFFFF" />
```

<TextView

```
android:id="@+id/rainTextView"
android:layout_width="186dp"
```

```
        android:layout_height="48dp"
        android:layout_below="@id/temperatureTextView"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="115dp"
        android:layout_marginTop="121dp"
        android:layout_marginEnd="110dp"
        android:text="Rain: "
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textColorHighlight="#FDFBFB" />
</RelativeLayout>
```

BACKEND JAVA CODING:-

Firestore.java:

```
package com.example.floodmonitoring;

import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.example.floodmonitoring.GaugeView;
import com.google.firebase.database.ValueEventListener;

public class Firestore extends AppCompatActivity {
```

```
private DatabaseReference sensorDataRef;

private GaugeView distanceGauge;

private GaugeView humidityGauge;

private GaugeView temperatureGauge;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_firebase);

    // Initialize Firebase

    FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();

    sensorDataRef = firebaseDatabase.getReference("sensor_data");

    // Get references to GaugeView widgets in your layout

    distanceGauge = findViewById(R.id.gaugeView);

    humidityGauge = findViewById(R.id.gaugeView3);

    temperatureGauge = findViewById(R.id.gaugeView4);

    // Set up a listener to retrieve data from Firebase

    sensorDataRef.addValueEventListener(new ValueEventListener() {

        @Override

        public void onDataChange(DataSnapshot dataSnapshot) {

            if (dataSnapshot.exists()) {

                // Retrieve values from the dataSnapshot

                double distance = dataSnapshot.child("Distance").getValue(Double.class);

                Integer humidity = dataSnapshot.child("Humidity").getValue(Integer.class);

                Integer temperature = dataSnapshot.child("Temperature").getValue(Integer.class);
```

```

        if (humidity != null && temperature != null) {

            // Update the GaugeView widgets with the retrieved values

            distanceGauge.setValue((int) distance);

            humidityGauge.setValue(humidity);

            temperatureGauge.setValue(temperature);

        } else {

        }

    }

}

@Override

public void onCancelled(DatabaseError databaseError) {

    // Handle database error

}

});

}

}

```

PROGRAM DESCRIPTION:-

This code is for an Android app that monitors and displays sensor data, particularly distance, humidity, and temperature, retrieved from a Firebase database. Let's break down the code step by step:

1. The app's package name is declared as "com.example.floodmonitoring."
2. The necessary Android libraries and classes are imported.
3. The `Firebase` class is defined, which extends `AppCompatActivity`, indicating that this is an Android activity.
4. Inside the `Firebase` class, several class-level variables are declared:
 - `sensorDataRef`: This variable is used to reference the Firebase database location where sensor data is stored.

- `distanceGauge`, `humidityGauge`, and `temperatureGauge`: These variables represent widgets on the app's user interface called "GaugeView" and will be used to display sensor data.

WEATHER.JAVA:-

```
package com.example.floodmonitoring;
```

```
import android.os.AsyncTask;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.ImageView;
```

```
import android.widget.TextView;
```

```
import android.widget.Toast;
```

```
import org.json.JSONObject;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
public class Weather extends AppCompatActivity {
```

```
    private EditText locationEditText;
```

```
    private Button getWeatherButton;
```

```
    private TextView locationTextView;
```

```
    private TextView temperatureTextView;
```

```
    private TextView rainTextView;
```

```
    private ImageView weatherIcon;
```

```
private String apiKey = "b141741991279045db84b225a342d85e"; // Replace with your
OpenWeatherMap API key
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_weather);
```

```
    locationEditText = findViewById(R.id.locationEditText);
```

```
    getWeatherButton = findViewById(R.id.getWeatherButton);
```

```
    locationTextView = findViewById(R.id.locationTextView);
```

```
    temperatureTextView = findViewById(R.id.temperatureTextView);
```

```
    rainTextView = findViewById(R.id.rainTextView);
```

```
    getWeatherButton.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
```

```
            String location = locationEditText.getText().toString();
```

```
            // Fetch weather data for the entered location
```

```
            new WeatherTask().execute(location);
```

```
        }
```

```
    });
```

```
}
```

```
class WeatherTask extends AsyncTask<String, Void, String> {
```

```
    @Override
```

```
    protected String doInBackground(String... params) {
```

```
        if (params.length != 1) {
```

```
            return "Invalid parameters"; // Handle invalid input
```

```
        }
```

```

String location = params[0];

String result = "";

try {
    String url = "http://api.openweathermap.org/data/2.5/weather?q=" + location +
"&appid=" + apiKey;

    URL apiURL = new URL(url);

    HttpURLConnection connection = (HttpURLConnection) apiURL.openConnection();

    InputStream inputStream = connection.getInputStream();

    InputStreamReader inputStreamReader = new InputStreamReader(inputStream);

    int data = inputStreamReader.read();
    while (data != -1) {
        char current = (char) data;

        result += current;

        data = inputStreamReader.read();
    }

    return result;
} catch (Exception e) {
    e.printStackTrace();

    return null;
}

@Override
protected void onPostExecute(String s) {
    super.onPostExecute(s);

```

```

try {
    JSONObject jsonObject = new JSONObject(s);

    String city = jsonObject.getString("name");

    String temperature = jsonObject.getJSONObject("main").getString("temp");

    // Extract rain information

    JSONObject rainData = jsonObject.optJSONObject("rain");

    String rainInfo = (rainData != null) ? rainData.toString() : "No rain data available";

    // Update the UI

    locationTextView.setText(city);

    temperatureTextView.setText(temperature + "°C");

    rainTextView.setText("Rain: " + rainInfo);

    // Load and display the weather icon (you may need to use a library like Picasso or Glide)

    // Set the weather icon based on the "icon" value (e.g., "01d" for clear sky)

    // Replace "R.drawable.icon01d" with the correct resource ID for your weather icons

    // weatherIcon.setImageResource(R.drawable.icon01d);

} catch (Exception e) {

    e.printStackTrace();

    Toast.makeText(Weather.this, "Error fetching weather data",
Toast.LENGTH_SHORT).show();

    }

}

}

```

Conclusion:

A flood monitoring and early warning system is vital project that can save lives and reduce property damage. this project outline provides a framework for developing such a system, but it's important to tailor it to the specific needs and resources of the target region or community.