School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

A
DISSERTATION REPORT ON

"WEATHER APP"

Submitted to
CMR University School Of Engineering Technology, Bagalur
for the partial fulfillment of the Requirement for the Award of the Degree of

B.TECH
IN
COMPUTER SCIENCE

Submitted by:
SARANYA.T
(REG.NO : 16UG08047)

Under the Guidance Of

Prof. PALLAVI

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING CMR
UNIVERSITY BAGALUR 2018-2019

School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

*Certified that the project work entitled "**WEATHER APP** " carried out by Ms.**SARANYA.T**, REG.NO **16UG08047** in partial fulfillment for the award of Bachelor of Technology in **COMPUTER SCIENCE** of the CMR University, Bagalur during the year 2018-19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

Prof.PALLAVi	***Prof.SMITHA RAO***	***DR.NAGRAJM.K***
Signature of the Guide	*Signature of the HOD*	*Signature of the Dean*

External Viva

Name of the examiners	**Signature with date**
1	
2	

DECLARATION

*I, SARANYA.T**,** student of CMR university school of engineering and technology, bagalur hearby declare that the dissertation entitled **"WEATHER APP"** embodies the report of my project carried out independently by me during fifth semester of **B.TECH in Computer science,** under the supervision and guidance of **Prof. PALLAVI** Department of Computer Science and Engineering and this work has been submitted for the partial fulfillment of the requirements for the award of the B.Tech degree.*

I have not submitted the matter embodies to any other university or institution for the award of other degree.

Date :

Place :

SARANYA.T

(16UG08047)

SIGNATURE OF STUDENT

ACKNOWLEDGEMENT

*I express my foremost gratitude to **Dr.NAGRAJ M.K**, Dean for his constant support.*

*I express my foremost gratitude to **Prof.SMITHA RAO**, HOD, Dept of Computer Science & Engineering, CMR University, Bagalur for her inspiration, direction and discussions in successful completion of this dissertation.*

*I express my foremost gratitude to my guide **Prof.PALLAVI**, Dept of Computer Science & Engineering, CMR University, Bagalur for his/her inspiration, adroit guidance, constant supervision, direction and discussions in successful completion of this dissertation.*

*My sincere thanks to all teaching and non-Teaching Staff of **Computer Science & Engineering Department**for all the facilities provided, without which, I could not have progressed with my work. Thanks to my parents who have been a great source of strength in the completion of this dissertation.*

SARANYA.T

(USN: 16UG08047)

ABSTRACT

Accuweather is one of the more solid weather apps. It features the basics, including extended forecasts, hourly forecasts, and the like. Other features include radar, Android Wear support, and more. It also includes a Minute Cast feature. It predicts rain on a minute-by-minute basis. The app itself looks pretty good. The widgets are serviceable. It's one of the better all-around weather apps. Weather is one of the more unique weather apps. It's quite flashy. You can move your finger across the UH to see the weather at any given point in the day. It also covers the basics fairly well. It's not as powerful as some weather apps. However, simplicity has its benefits as well. This is a great weather app for those who need something simple but also still looks good.

Use Accuweather for the same reason you should: it's dependable and damn accurate. Minute-cast is scarily accurate, and it's hard to leave Accuweather for another weather app without it. Hyper-local, hyper-accurate forecasting is something of a misnomer, but it's something that's invaluable when it works. Accweather has Minute-cast, which gives you a to-the-minute timeline for the next two hours predicting when rain will start, stop, and how hard it will be. In places where rain can start and stop seemingly at random—like wonderful,thunder-proneOrlando.

TABLE OF CONTENTS

CHAPTER 1
PREAMBLE

1.1 Introduction

Weather is what is happening in the atmosphere at any time or short period of time. Weather conditions can change suddenly. Today may be warm and sunny, tomorrow may be cool and cloudy. Weather conditions include Clouds, rain, snow, sleet, hail, fog, mist, sunshine, wind, temperature and thunderstorms.

Weather is driven by the heat stored in the Earth's atmosphere, which comes from energy from the Sun. When heat is moved around the Earth's surface and in the atmosphere because of differences in temperature between places, this makes winds. Winds form part of larger weather systems, the most powerful of which is the hurricane. Other weather features like the thunderstorms also develop because of the movement of heat in the atmosphere. Some thunderstorms in the United States give birth to tornodes.

Like energy, water is moved between the Earth and the atmosphere. The Earth's watercycles plays an important role in the development of many weather features like dews,fogs,clouds and rain.

Scientists measures the weather so they can forecast it. This involves plotting weather information on special charts. Weather radar and satellites are now also used to help predict the weather.

Weather forecastingis the application of science and technology to predict the conditions of the atmosphere for a given location and time. People have attempted to predict the weather normally and formally since the 19thcentury. Weather forecasts are made by collecting quantitative data bout the current state of the atmosphere at a given place and using meterology to project how the atmosphere will change.

1.2 Literature Survey

Agrawal et al. [1]: explained the phenomena for time series regression models for forecasting the yield of rice in Raipur district on weekly data using weather parameters [1]. In [2] the author Kuo and Sun, (1993) was used to Associate in having intervention model for average10 days stream flow forecast and synthesis that was investigated by to

effect the extraordinary phenomena caused by typhoons and different serious irregularities of the weather of the Tanshui geographical area in Taiwan. In [3] Chiew et al, (1993) conducted a comparison of six rainfall-runoff modeling approaches to pretend daily, monthly and annual flows in eight tolerant catchments.

Chatfield [2]: Montgomery and Lynwood 1996). Several authors have discussed the fuzziness associated with the weather systems. In [7] Chaotic features are associated with the atmospheric phenomena also have fascinated the attention of the modern scientists.

Sivakumar et al.[3]: At present, the valuation of the nature and causes of seasonal climate variability is still formation. Since, it is a complicated phenomenon that includes many specialized fields of know-how to work for weather prediction (Guhathakurata, 2006); therefore, in the field of meteorology all assumptions are to be taken in the visage of uncertainty connected with local of and global climatic variables.

H.F. Blanford[4]: who had established the India Meteorological Department in 1875, issued the first seasonal forecast of Indian monsoon rainfall in 1884. Later, in the early part of the 20th century, Sir Gilbert Walker initiated extensive studies of global teleconnections which led him to the discovery of Southern Oscillation. Walker introduced, for the first time, the concept of correlation for long-range forecasting of the Asian summer monsoon and his findings are relevant even today.

1.3 Problem statement

Creating a weather app on Android is easy and a damn cool thing you can use by yourself. You don’t need to setup a weather station in each city on each country to get weather information in your app. You can get those information including Temperature,Pressure,Humidity,Weather status by using weather Api.

1.4Objective

To setup the Weather app for detecting temperature and pressure with javascript and xml by adding new configuration then app presents which exists.Its innovative weather can be identified with open weather map.

1.5 Methodology

If we want to go to a new place, then we cannot check the weather through our existing application in that place .We have to check through search engines like google. The existing system is not portable(hard to use). We can check the weather reports (i.e., prediction reports) through our proposed system. We can check the weather at other places through our proposed system.Android mobile with a minimum version.The processor is not less than 500MHZ.RAM > 170mb.SD card with a minimum of 512 MB. Resolution is not less than 480*800pixs.This system is portable (easy to use).

CHAPTER 2

GENERAL ASPECTS AND TECHNOLOGY

Weather is driven by the heat stored in the Earth's atmosphere, which comes from energy from the Sun. When heat is moved around the Earth's surface and in the atmosphere because of differences in temperature between places, this makes winds. Winds form part of larger weather systems, the most powerful of which is the hurricane. Other weather features like the thunderstorms also develop because of the movement of heat in the atmosphere. Some thunderstorms in the United States give birth to tornodes.

Use Accuweather for the same reason you should: it's dependable and damn accurate. Minute-cast is scarily accurate, and it's hard to leave Accuweather for another weather app without it. Hyper-local, hyper-accurate forecasting is something of a misnomer, but it's something that's invaluable when it works. Accuweather has Minute-cast, which gives you a to-the-minute timeline for the next two hours predicting when rain will start, stop, and how hard it will be. In places where rain can start and stop seemingly at random — like wonderful, thunder-prone Orlando.

CHAPTER 3

SYSTEM SPECIFICATION AND DESIGN

HARDWARE REQUIREMENTS:

In the existing system, we don't have mobile applications to check the weather conditions. If we go some new place, then we have to check the weather conditions instantly. So, to check instantly, we need a mobile application because, mobile is part of our life presently. Anyone can handle the mobile device easily

1) Android mobile with a minimum version 2.2.

2) The processor is not less than 500MHZ.

3) RAM > 170mb.

4) SD card with a minimum of 512 MB.

5) Resolution is not less than 480*800pixs.

SOFTWARE REQUIREMENTS:

1) Android Studio.

2) JavaScript.

3) XML

CHAPTER 4

IMPLEMENTATION

1.First of all create a Weather API from open weather map.

2.Create a new project in Android Studio in a normal way.

3.Copy**weathericons-regular-webfont.ttf**o your

project's**src/main/assets/fonts** directory. You can download the font from **.** Or you can download from our website and extract the zip file.

4. Open your **AndroidManifest.xml** file and add **internet connect permission**. Your manifest file will look like this-

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.androstock.myweatherapp">
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
```

```
android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>
```

5.Open your**activity_main.xml** file. Use a RelativeLayout to arrange the text views. You can adjust the textSize to suit various devices.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

android:layout_width="match_parent"

android:layout_height="match_parent"

android:background="#3F51B5"

android:padding="20dp">

<TextView

android:id="@+id/city_field"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignParentTop="true"
```

```
android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceLarge" />


<TextView

android:id="@+id/updated_field"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/city_field"

android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceMedium"

android:textSize="13sp" />

<TextView

android:id="@+id/selectCity"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Change City"

android:textStyle="italic"

android:textSize="11dp"

android:textColor="@color/colorAccent"
```

```
android:layout_below="@+id/updated_field"

android:layout_centerHorizontal="true"/>

<TextView

android:id="@+id/weather_icon"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_centerVertical="true"

android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textSize="90sp"

<TextView

android:id="@+id/current_temperature_field"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignParentBottom="true"

android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textSize="50sp" />

<TextView
```

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

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/weather_icon"

android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceMedium"

<TextView

android:id="@+id/humidity_field"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/details_field"

android:layout_centerHorizontal="true"

android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceMedium"

<TextView

android:id="@+id/pressure_field"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/humidity_field"

android:layout_centerHorizontal="true"
```

```
android:textColor="#FFFFFF"

android:textAppearance="?android:attr/textAppearanceMedium”

<ProgressBar

android:id="@+id/loader"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_centerInParent="true"/>

</RelativeLayout>
```

Finally we are heading towards the Java coding.

6. Now we are going to create a Java class file named **Function.java.** Here we will write all our functions so that we can use them easily from **MainActiviy.java** letter.

We use the **HttpURLConnection** class to make the remote request. We use a **BufferedReader** to read the API's response into a **StringBuffer**. When we have the complete response, we convert it to a **JSONObject** object.

```
package com.androstock.myweatherapp;

import android.content.Context;

import android.net.ConnectivityManager;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;
```

```
import java.util.Date;

public class Function {

// Project Created by Ferdousur Rahman Shajib

// www.androstock.com

public static boolean isNetworkAvailable(Context context)

{

return ((ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE)).getActiveNetworkInfo() !
= null;

}

public static String excuteGet(String targetURL)

{

URL url;

HttpURLConnection connection = null;

try {

//Create connection

url = new URL(targetURL);

connection = (HttpURLConnection)url.openConnection();

connection.setRequestProperty("content-type", "application/json;  charset=utf-8");

connection.setRequestProperty("Content-Language", "en-US");

connection.setUseCaches (false);

connection.setDoInput(true);
```

```
connection.setDoOutput(false);

InputStream is;

int status = connection.getResponseCode();

if (status != HttpURLConnection.HTTP_OK)

is = connection.getErrorStream();

else

is = connection.getInputStream();

BufferedReader rd = new BufferedReader(new InputStreamReader(is));

String line;

StringBuffer response = new StringBuffer();

while((line = rd.readLine()) != null) {

response.append(line);

response.append('\r');

}

rd.close();

return response.toString();

} catch (Exception e) {

return null;

} finally {

if(connection != null) {

connection.disconnect();
```

```
}

}

}

public static String setWeatherIcon(int actualId, long sunrise, long sunset){

int id = actualId / 100;

String icon = "";

if(actualId == 800){

long currentTime = new Date().getTime();

if(currentTime>=sunrise && currentTime<sunset) {

icon = "&#xf00d;";

} else {

icon = "&#xf02e;";

}

} else {

switch(id) {

case 2 : icon = "&#xf01e;";

break;

case 3 : icon = "&#xf01c;";

break;

case 7 : icon = "&#xf014;";

break;
```

```
case 8 : icon = "&#xf013;";

break;

case 6 : icon = "&#xf01b;";

break;

case 5 : icon = "&#xf019;";

break;

}

}

return icon;

}}
```

6. Now time to move towards the **MainActivity.java** class. We are going to make the activity fullscreen. So add the following line before your setContentView().

```
getSupportActionBar().hide();
```

Your **MainActivity.java** will look like-

```
package com.androstock.myweatherapp;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.graphics.Typeface;

import android.os.AsyncTask;
```

```
import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.text.Html;

import android.view.View;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import org.json.JSONException;

import org.json.JSONObject;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

// Project Created by Ferdousur Rahman Shajib

// www.androstock.com

TextView selectCity, cityField, detailsField, currentTemperatureField, humidity_field,
pressure_field, weatherIcon, updatedField;

ProgressBar loader;

Typeface weatherFont;
```

```
String city = "Dhaka, BD";

/* Please Put your API KEY here */

String OPEN_WEATHER_MAP_API = "cbfdb21fa1793c10b14b6b6d00fbef03";

/* Please Put your API KEY here */

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

getSupportActionBar().hide();

setContentView(R.layout.activity_main);

loader = (ProgressBar) findViewById(R.id.loader);

selectCity = (TextView) findViewById(R.id.selectCity);

cityField = (TextView) findViewById(R.id.city_field);

updatedField = (TextView) findViewById(R.id.updated_field);

detailsField = (TextView) findViewById(R.id.details_field);

currentTemperatureField = (TextView) findViewById(R.id.current_temperature_field);

humidity_field = (TextView) findViewById(R.id.humidity_field);

pressure_field = (TextView) findViewById(R.id.pressure_field);

weatherIcon = (TextView) findViewById(R.id.weather_icon);

weatherFont = Typeface.createFromAsset(getAssets(), "fonts/weathericons-regular-
webfont.ttf");

weatherIcon.setTypeface(weatherFont);
```

```
taskLoadUp(city);
selectCity.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.this);
alertDialog.setTitle("Change City");
final EditText input = new EditText(MainActivity.this);
input.setText(city);
LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(
LinearLayout.LayoutParams.MATCH_PARENT,
LinearLayout.LayoutParams.MATCH_PARENT);
input.setLayoutParams(lp);
alertDialog.setView(input);

alertDialog.setPositiveButton("Change",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
city = input.getText().toString();
taskLoadUp(city);
}
});
```

```
alertDialog.setNegativeButton("Cancel",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
dialog.cancel();
}
});
alertDialog.show();
}
});
public void taskLoadUp(String query) {
if (Function.isNetworkAvailable(getApplicationContext())) {
DownloadWeather task = new DownloadWeather();
task.execute(query);
} else {
Toast.makeText(getApplicationContext(), "No Internet Connection",
Toast.LENGTH_LONG).show();
}
}
class DownloadWeather extends AsyncTask < String, Void, String > {
@Override
protected void onPreExecute() {
```

```
super.onPreExecute();
loader.setVisibility(View.VISIBLE);
}
protected String doInBackground(String...args) {
String xml = Function.excuteGet("http://api.openweathermap.org/data/2.5/weather?q=" +
args[0] +
"&units=metric&appid=" + OPEN_WEATHER_MAP_API);
return xml;
}
@Override
protected void onPostExecute(String xml) {
try {
JSONObject json = new JSONObject(xml);
if (json != null) {
JSONObject details = json.getJSONArray("weather").getJSONObject(0);
JSONObject main = json.getJSONObject("main");
DateFormat df = DateFormat.getDateTimeInstance();
cityField.setText(json.getString("name").toUpperCase(Locale.US) + ", " +
json.getJSONObject("sys").getString("country"));
detailsField.setText(details.getString("description").toUpperCase(Locale.US));
currentTemperatureField.setText(String.format("%.2f", main.getDouble("temp")) + "°");
humidity_field.setText("Humidity: " + main.getString("humidity") + "%");
```

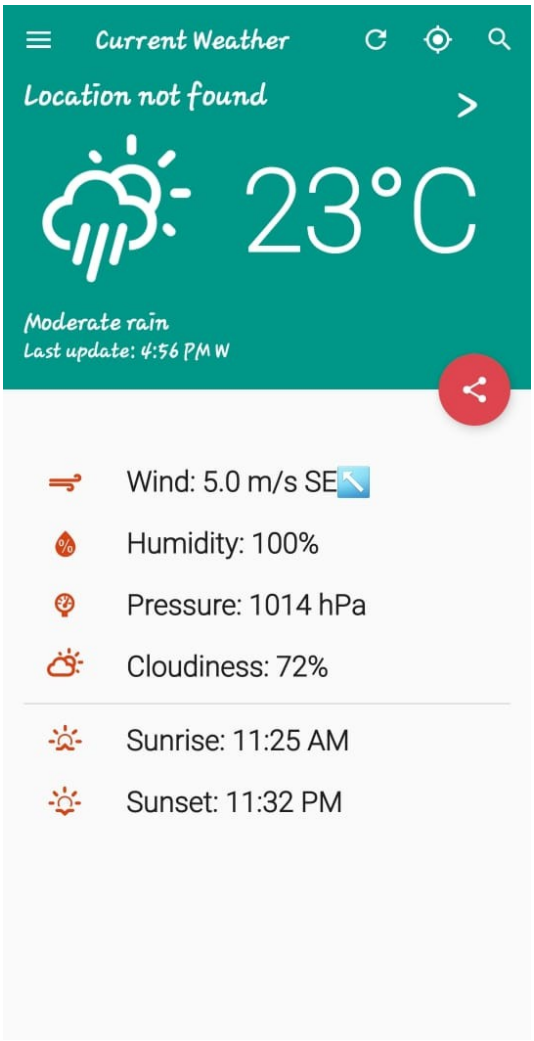
```
pressure_field.setText("Pressure: " + main.getString("pressure") + " hPa");

updatedField.setText(df.format(new Date(json.getLong("dt") * 1000)));

weatherIcon.setText(Html.fromHtml(Function.setWeatherIcon(details.getInt("id"),

json.getJSONObject("sys").getLong("sunrise") * 1000

json.getJSONObject("sys").getLong("sunset") * 1000)));

loader.setVisibility(View.GONE);

}

} catch (JSONException e) {

Toast.makeText(getApplicationContext(),

Toast.LENGTH_SHORT).show();

}

}

}}
```

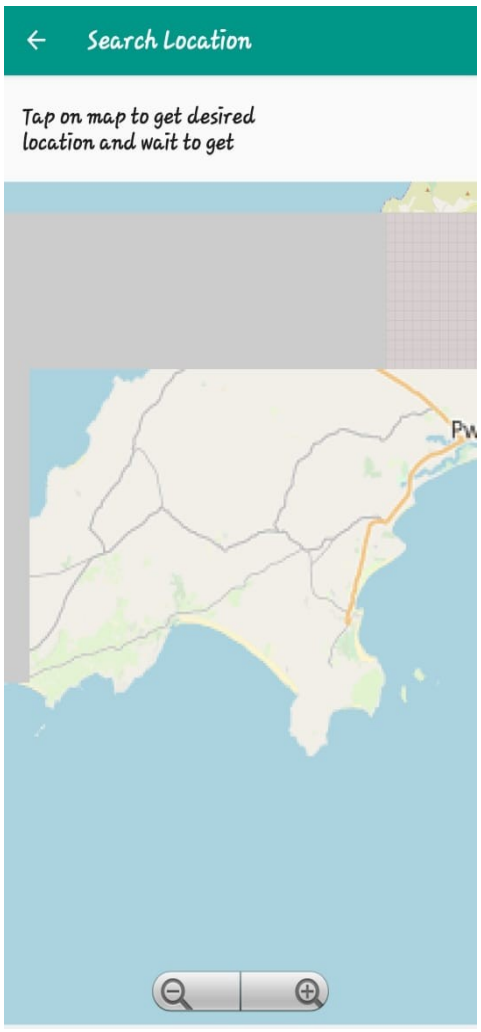
CHAPTER 5

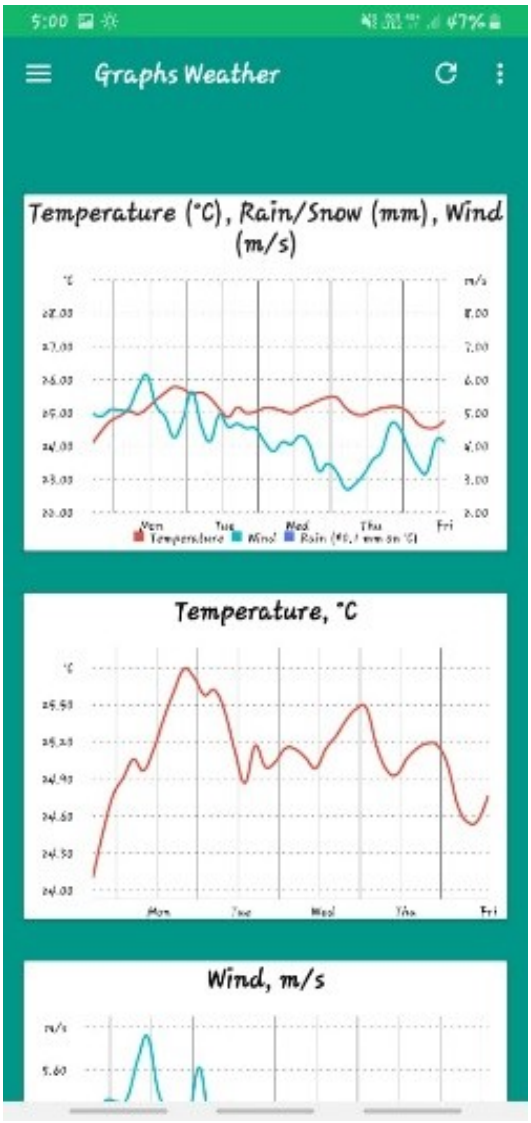
RESULTS AND DISCUSSIONS

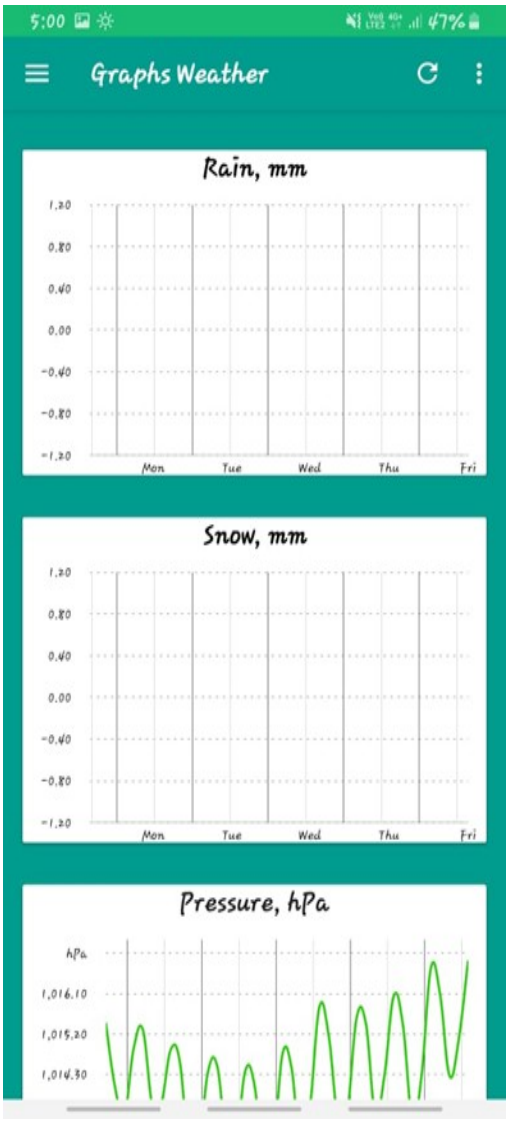
Now we can find the weather conditions using this waether app to find temperature and pressure.

We can implement this in forall the climatic conditions to process the specification of weather application.






5:00
47%
Graphs Weather
Temperature (°C), Rain/Snow (mm), Wind (m/s)
Mon
Tue
Wed
Thu
Fri
Temperature
Wind
Temperature, °C
Wind, m/s


5:00
47%
Graphs Weather
Rain, mm
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Mon
Tue
Wed
Thu
Fri
Snow, mm
Pressure, hPa
hPa
1,016.10
1,015.20
1,014.30

CONCLUSION

We may have a advantage of creating a own app for weather detecting for temperature and pressure.If we want to go to a new place, then we cannot check the weather through our existing application in that place .We have to check through search engines like google. The existing system is not portable(hard to use). We can check the weather reports (i.e., prediction reports) through our proposed system. We can check the weather at other places through our proposed system.Android mobile with a minimum version.The processor is not less than 500MHZ.RAM > 170mb.SD card with a minimum of 512 MB. Resolution is not less than 480*800pixs.

REFERENCES

1.https://inducesmile.com/android/create-a-beautiful-android-weather-app-using-openweathermap-api-and-volley-library-part-2

2. https://code.tutsplus.com/tutorials/create-a-weather-app-on-android--cms-21587

3. https://mediummm/@sasude9/sbasic-android-weather-app-6a7c0855caf4

4.https://www.javacodegeeks.com › Android › Android Core