**MINI PROJECT - I**

(2020-21)

WEATHER ADVISOR ANDROID APPLICATION

Final Report



Institute of Engineering & Technology

Made By: Palak Maheshwari

(181500442)

Under supervision of:

Mr. Mandeep Singh

(Technical Trainer)

Department of Computer Engineering & Applications

**Contents**

|  |  |
| --- | --- |
| **Abstract** | **03** |
| **1. Introduction** | **04** |
| 1.1 General Introduction to the topic | **04-05** |
| 1.2 Area of Computer Science | **06** |
| 1.3 Hardware and Software Requirements | **07** |
| **2. Problem definition** | **08** |
| **3. Objectives** | **09** |
| 4**. Implementation Details**  **5.Working Implementation**  **6. Screen Shots**  **7. References** | **10**  **11-12**  **13-30**  **31** |
|  |  |

**Abstract**

A huge change has occurred in the way people obtain weather information in the last few years and a large percentage of the population now get weather forecasts on their mobile phones. There is currently a wide range of smartphone weather apps available: in 2014, iTunes App Store alone offered 5043 active applications in the weather category. The rapid penetration of new broadcasting technologies strongly affects the way weather forecasts are communicated to, and used by, people. Portability, permanent connectivity and geolocalization allow location‐specific and time‐sensitive weather forecasts to be provided. This paper explores the main features emerging in the 39 most popular weather apps in the United States, United Kingdom and Italy, and focuses on the implications in the communication of uncertainty. The results show that even if the advances in mobile communication technologies could, in principle, improve the effectiveness of weather communication enormously, the expectations created around weather forecasts appear to be inconsistent with current forecasting capabilities, particularly with their inherent uncertainties in space and time, as well as in the nature of the predicted weather events

**Introduction**

**1.1 General Introduction to the topic**

A huge change has occurred in the way people obtain weather information in the last few years and a large percentage of the population now get weather forecasts on their mobile phones. In today's world weathering about your own location or about any other location is a must necessity. Here I'm proposing a weathering android application that might have some more stability in it and can tell accurately about the weather changing in the surrounding environment of a particular location. This all actions will be performed using Flutter and Android Studio. These Technologies really provide an amazing GUI Interface Always-on connectivity allows users to receive specific information or notifications and always reach updated information, accessing the data stream with immediacy. The push-functionality becomes critical, for example, to disseminate alerts and updated weather warnings, providing relevant and timely information in order to reduce vulnerability to weather hazards. These features, united with GPS (Global Positioning System) and location-enabled smartphones, allow new mobile technologies to provide location-specific and time-sensitive weather forecasts in support of decision making by various end users, thus accomplishing the ultimate goal.

# About Flutter : -

# Flutter is an open-source UI software Development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.

# Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS.

# The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and *Cupertino* widgets implement Apple's iOS Human interface guidelines

# About Android Studio: -

* Android Studio is the officialintegrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.
* Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version.

# How Android Studio Works: -

* Gradle-based build support.
* Android-specific refactoring and quick fixes
* Lint tools to catch performance, usability, version compatibility and other problems
* ProGaurd integration and app-signing capabilities
* Template-based wizards to create common Android designs and components.
* Support for building Android Wear apps.
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

# 1.2 Area of Computer Science

The user can obtain detailed information about the temporal range of interest by going to the secondary level step page. At this level, the elements are also wind (direction and speed), probability of precipitation and visibility. It can help farmers to get easy notifications about weather change regarding their crop climatic requirement. To provide more consistent and reliable weathering system.

# Hardware Requirements

**Software Specification**

* Technology Implemented : Android Studio
* Language Used : Java
* Database : openweathermap.org
* User Interface Design : Android Application
* Web Browser : Chrome

**Hardware Requirement**

* Processor : Intel CORE i3
* Operating System : Any Operation System
* RAM : 4 GB (8GB Recommended)
* Hardware System : Computer System
* Hard Disk : 1 TB

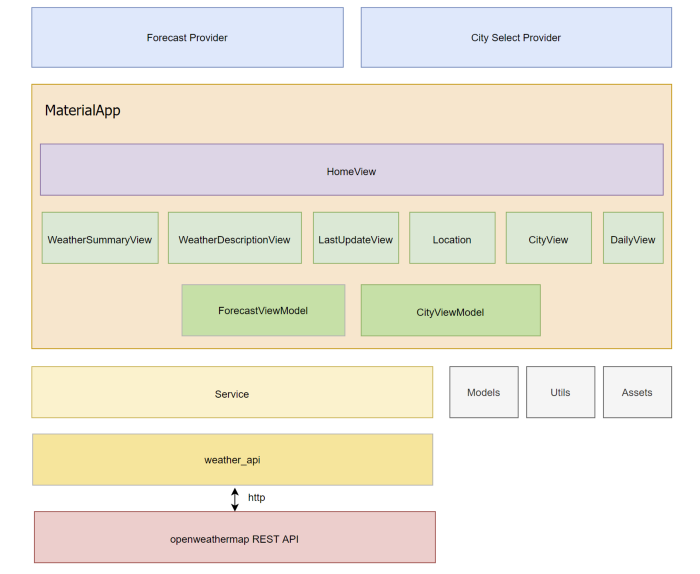
**Problem Solving**

With this project we will be aiming to Speed Up the application as it will take Less of the Internet Usage. To add Voice Implementation for visually impaired. To add Language Options so that anyone regardless to the particular language can use it.

# Objective

In order to reduce the difficulty level to provide utmost perfect temperate requirement according to their crop. It will be designed in such a way that anyone from agricultural activity.

# Implementation Details

****

**Working Implementation**

We have created many files as in implementation of this android application.

We mainly used 2 different types of programming languages that is Java and XML.

Here are the names of files use in this application:

1. MainActivity.java
2. WeatherConnect.java
3. CityPrefs.java
4. WeatherParser.java
5. Utils.java
6. Clouds.java
7. CurrentCondition.java
8. Locations.java
9. Snow.java
10. Temperature.java
11. Weather.java
12. Winds.java
13. AndroidManifest.xml
14. activity\_main.xml
15. string.xml
16. menu\_main.xml

**Working of the files:**

* **activity\_main.xml : This file is basically used to design the structural layout of the interface.**
* **menu\_main.xml : This file is used to add buttons to the action bar of application.**
* **string.xml : This file is to link activity\_main.xml and menu\_main.xml to provide the desired layout.**
* **AndroidManifest.xml : This file is to give the permission to internet linkage so that we can fetch our data from openweathermap.org .**
* **MainActivity.java : This File is to contain all the main programming to push data into our application. And to connect all the other java files.**

**We created few packages to place some java files under same package :**

1. **Util**
2. **Model**
3. **Data**

**Package Util contains a java file named Utils.java which contains all our static variables that are to be used as references.**

**Package model contains some java files that are named on different climatic conditions ( clouds, snow, winds, temperature, current condition, location, weather ) the and every class will hold the instance of this respective name.**

**Package data contains some important java files that are needed to pass the JSON objects in them.**

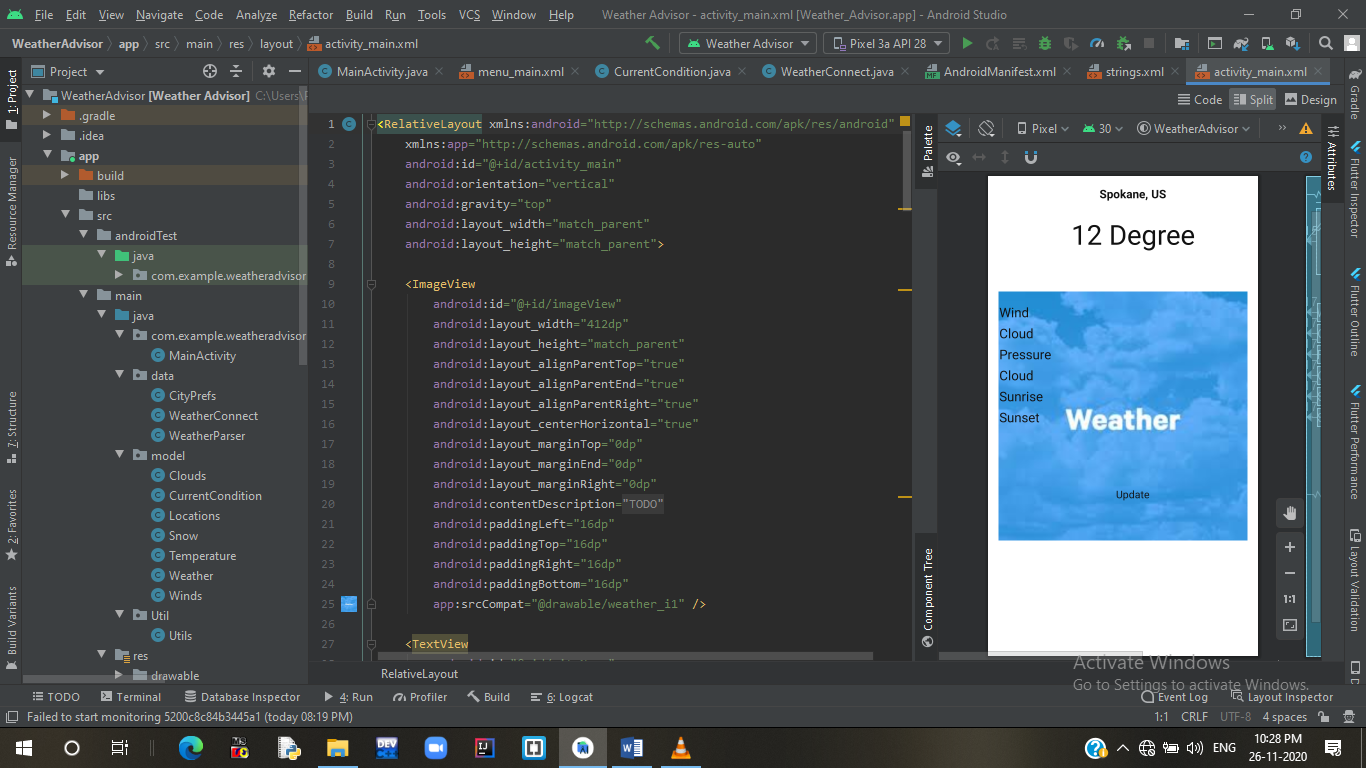
**WeatherParser.java**

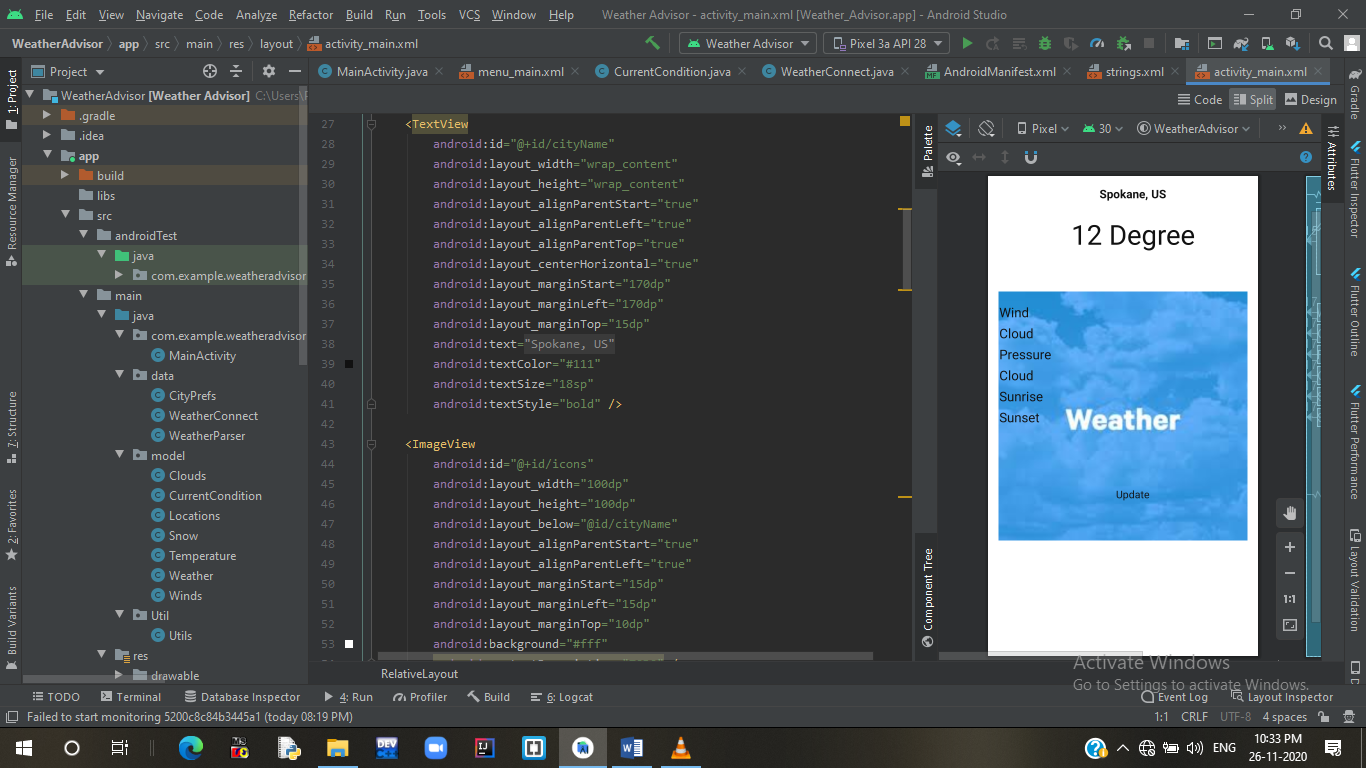
**WeatherConnect.java**

**CityPrefs.java**

**SCREENSHOTS**

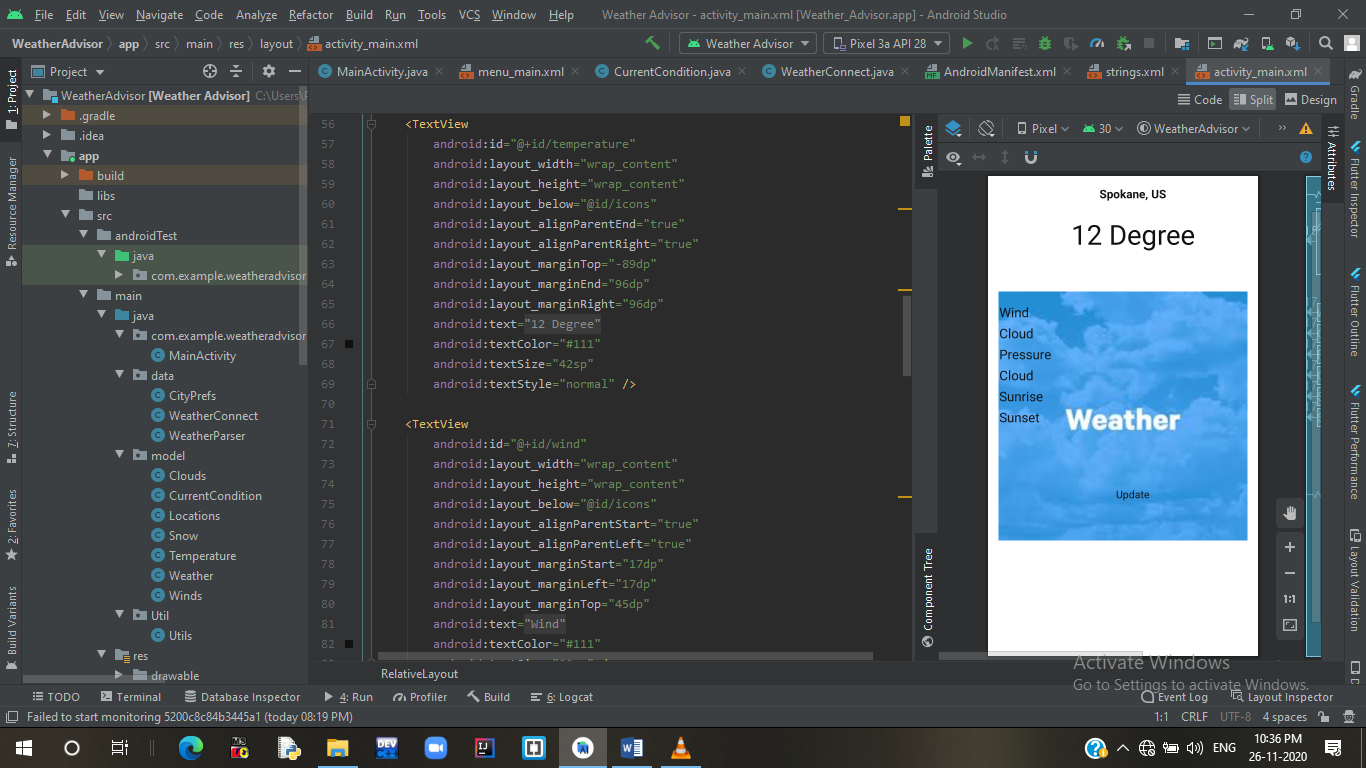
**File: activity\_main.xml**

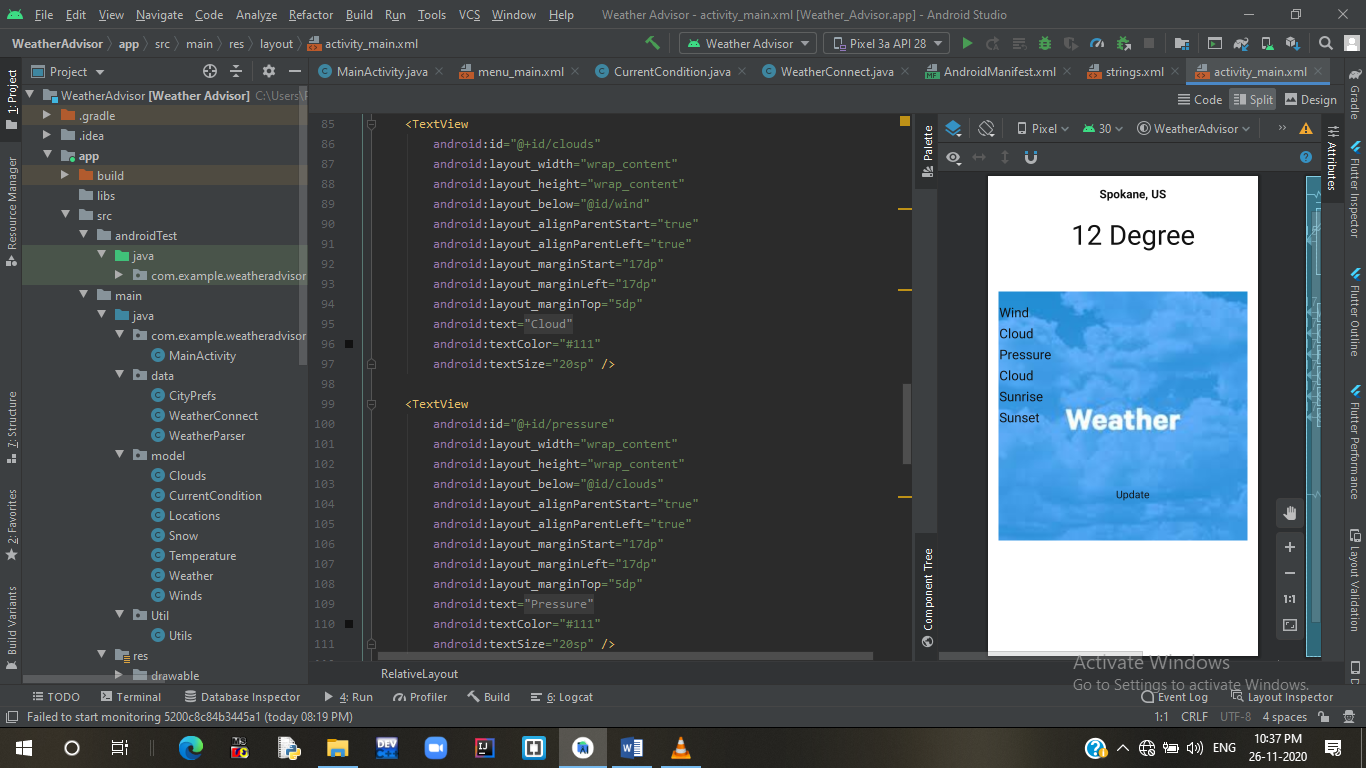


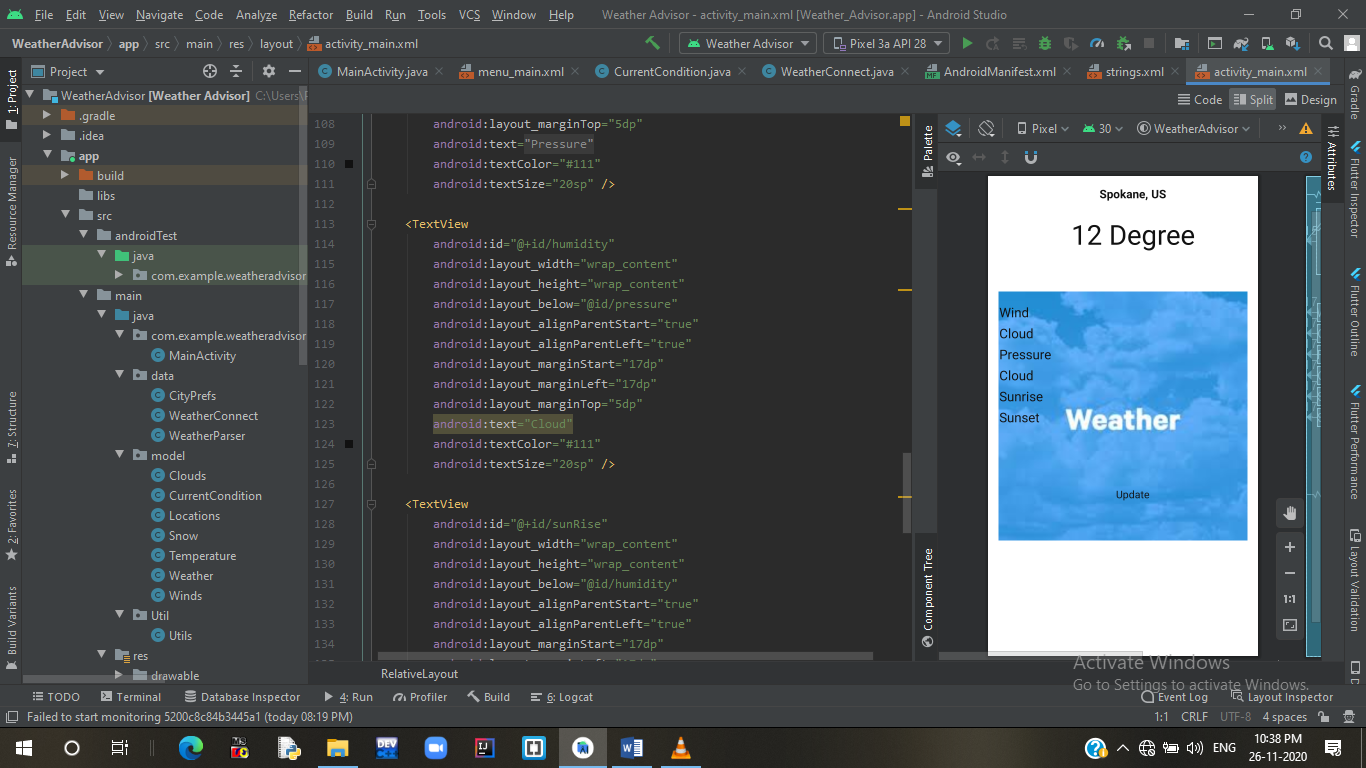


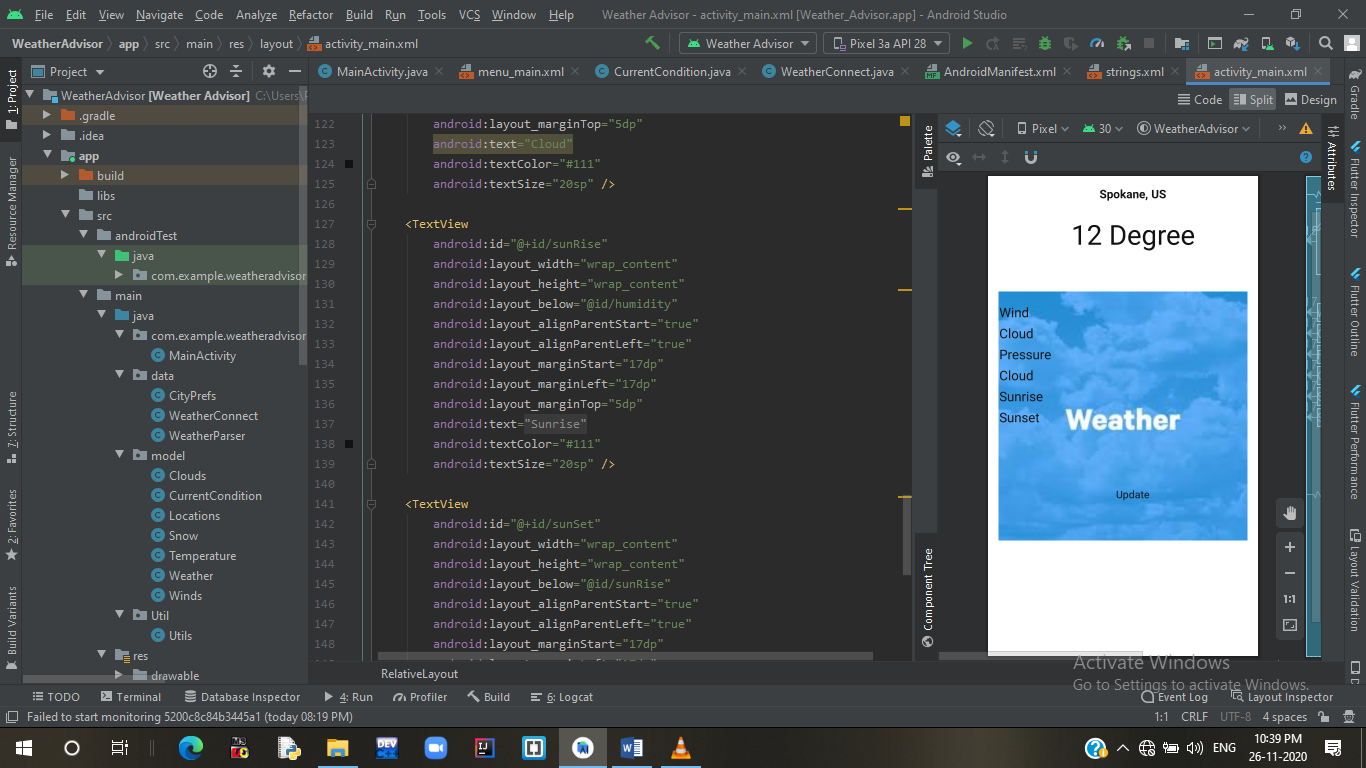
**File :AndroidManifest.xml**

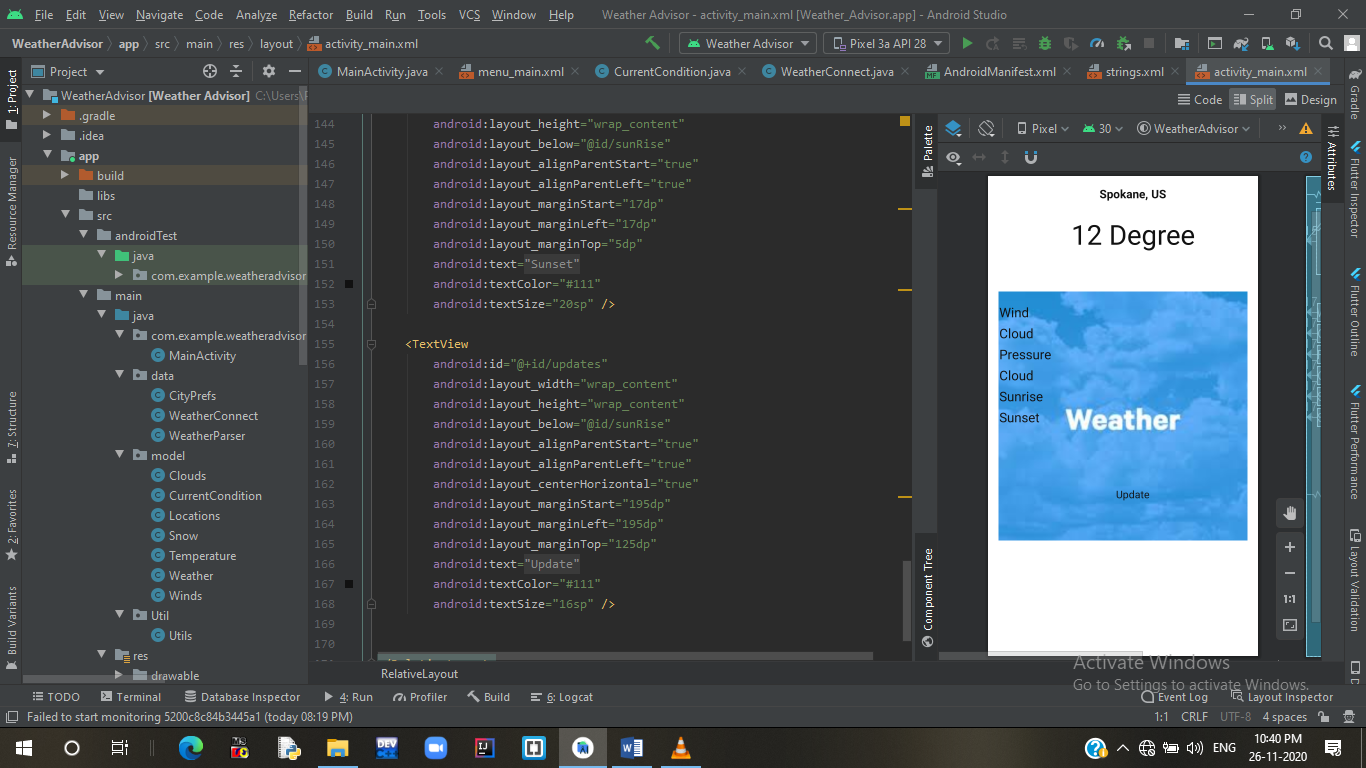
**References**



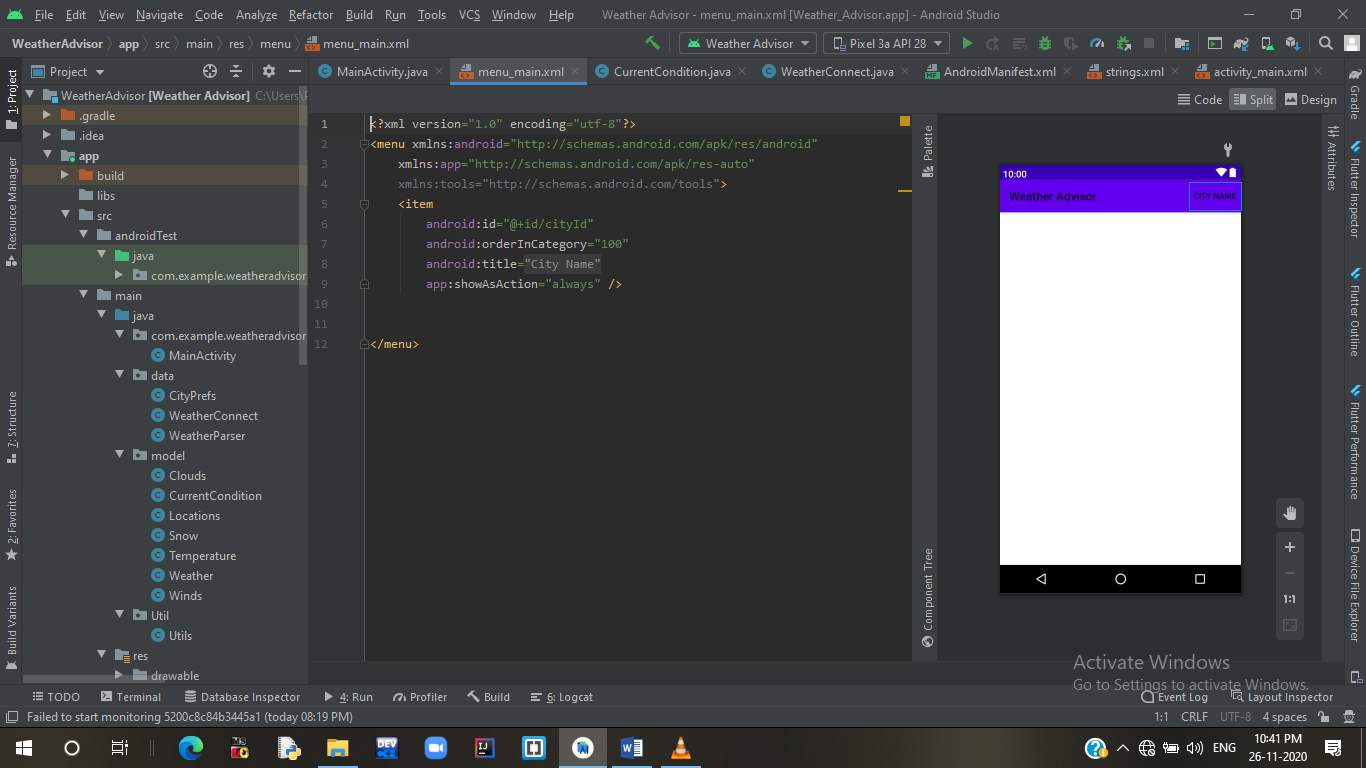




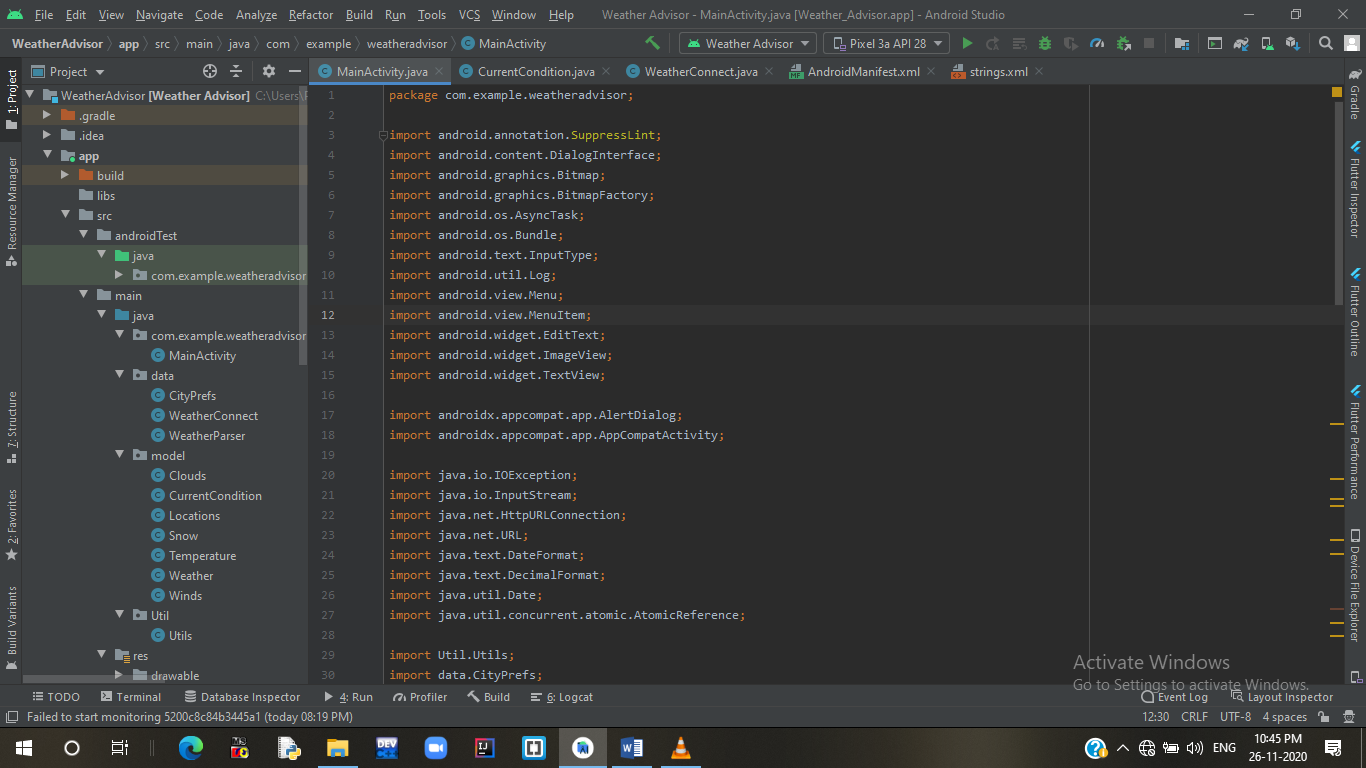


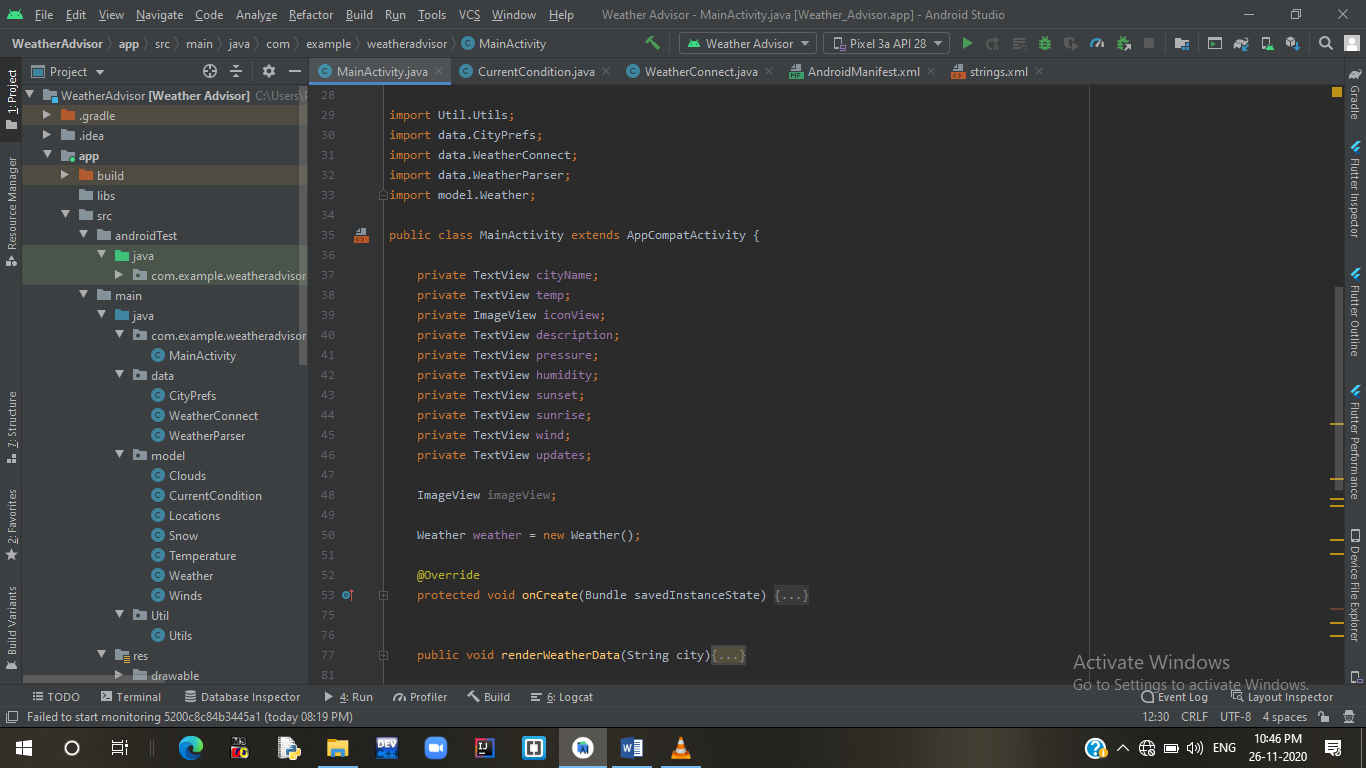


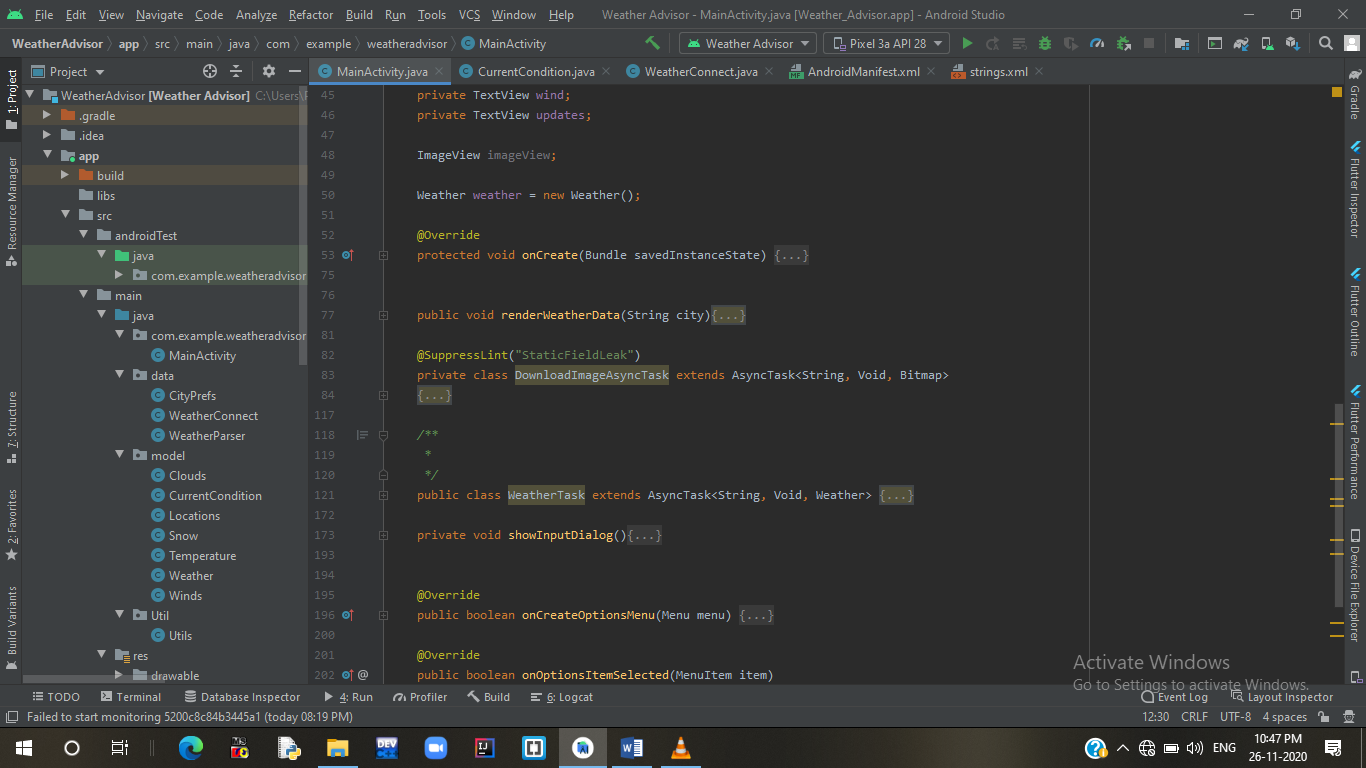
File : menu\_main.xml

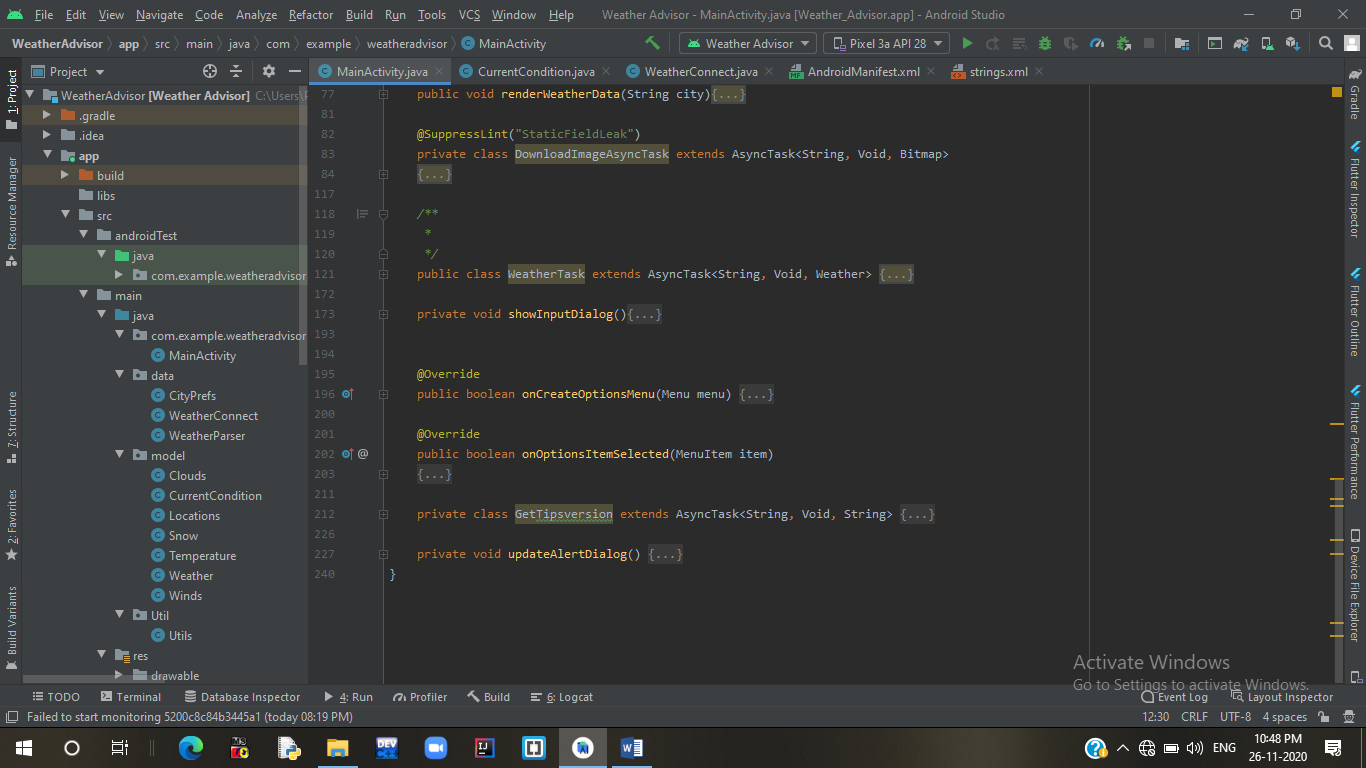


File : MainActivity.java

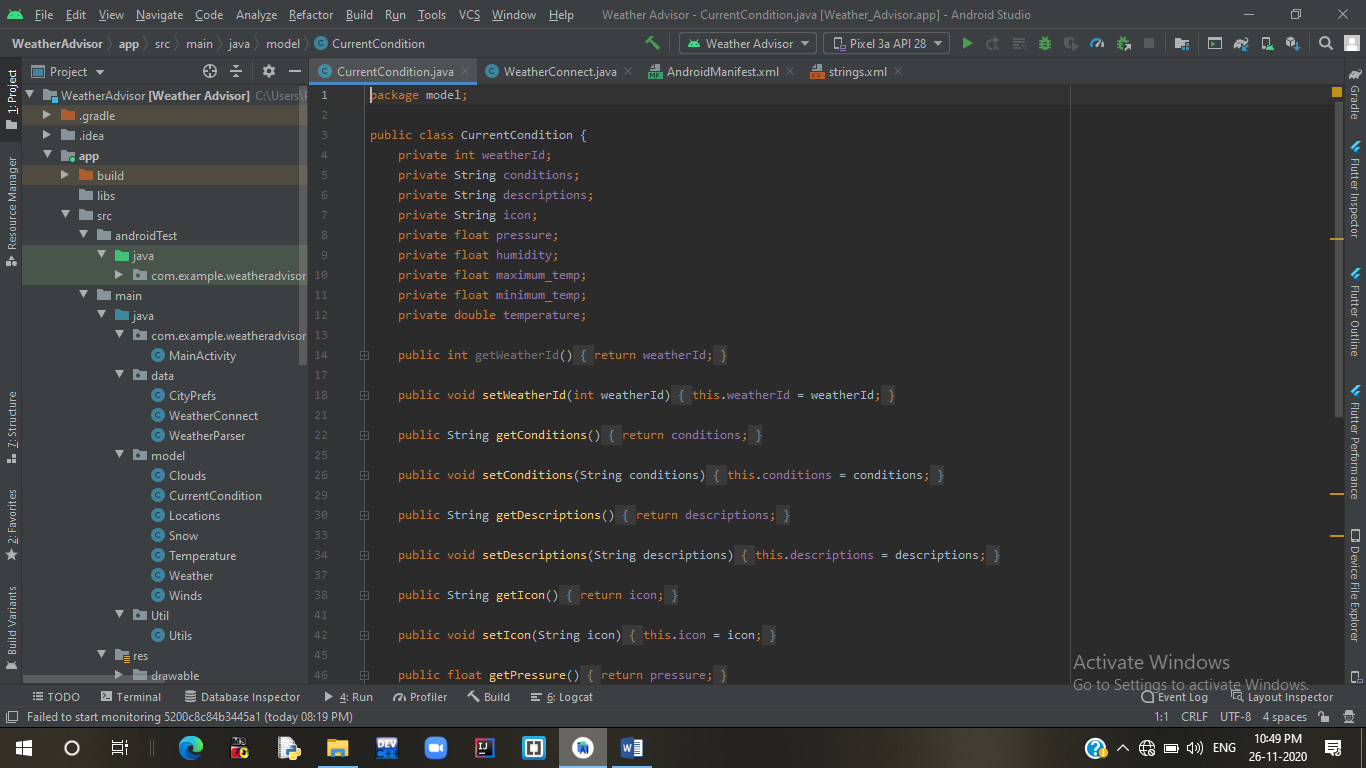


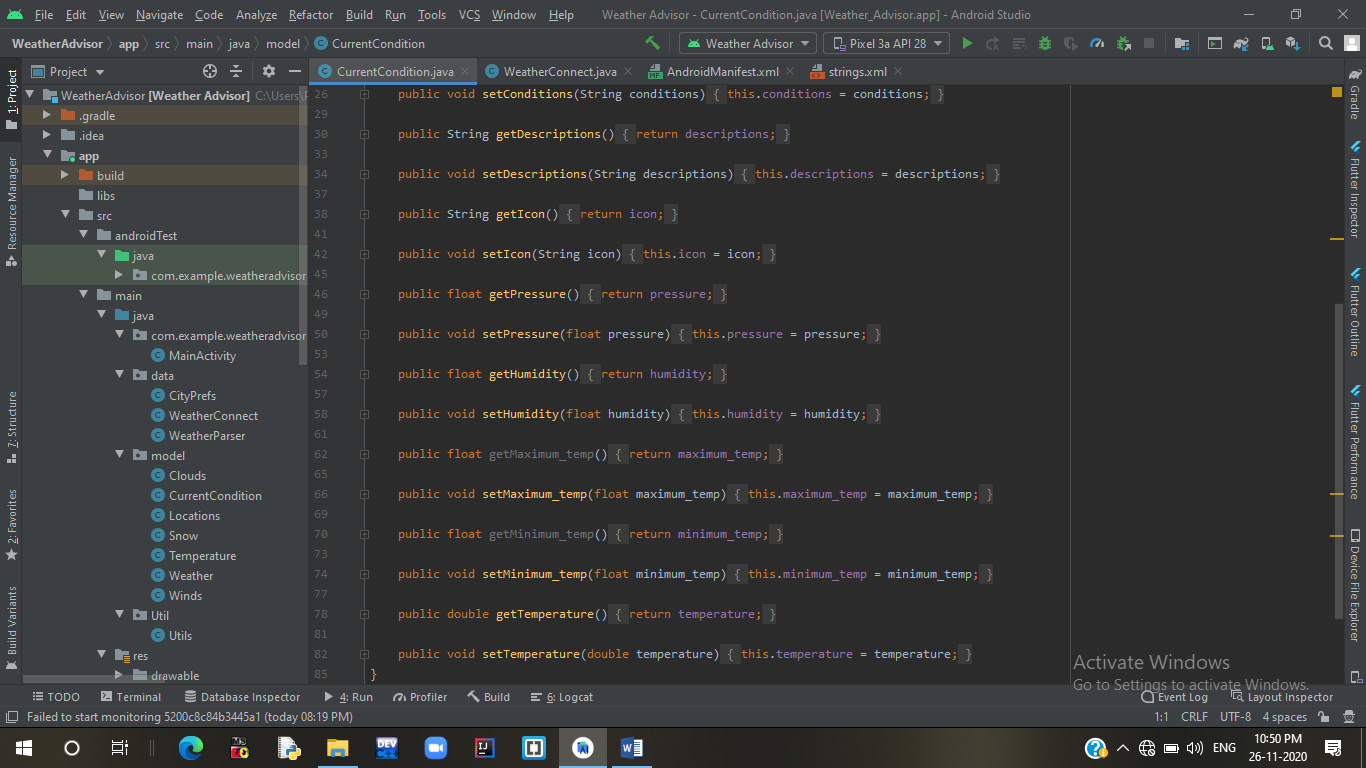




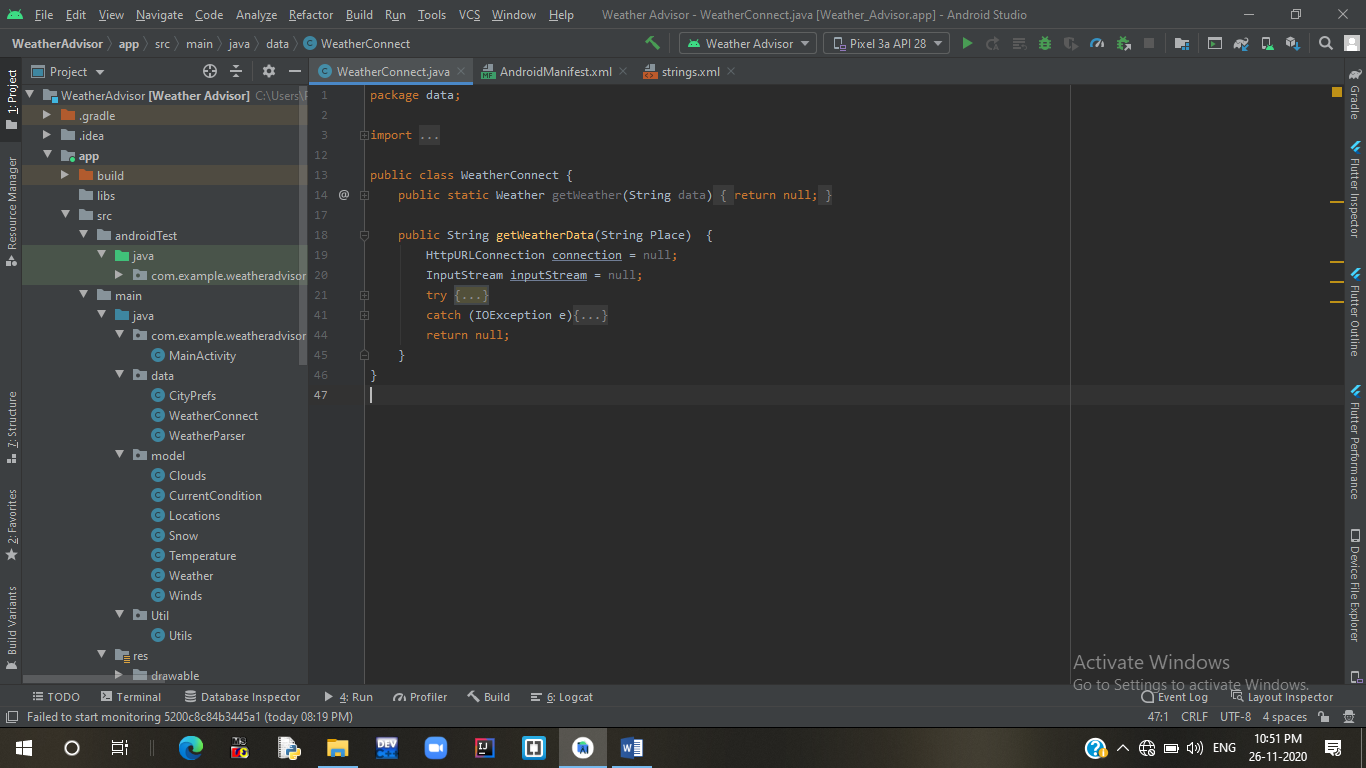


File : CurrentCondition.java

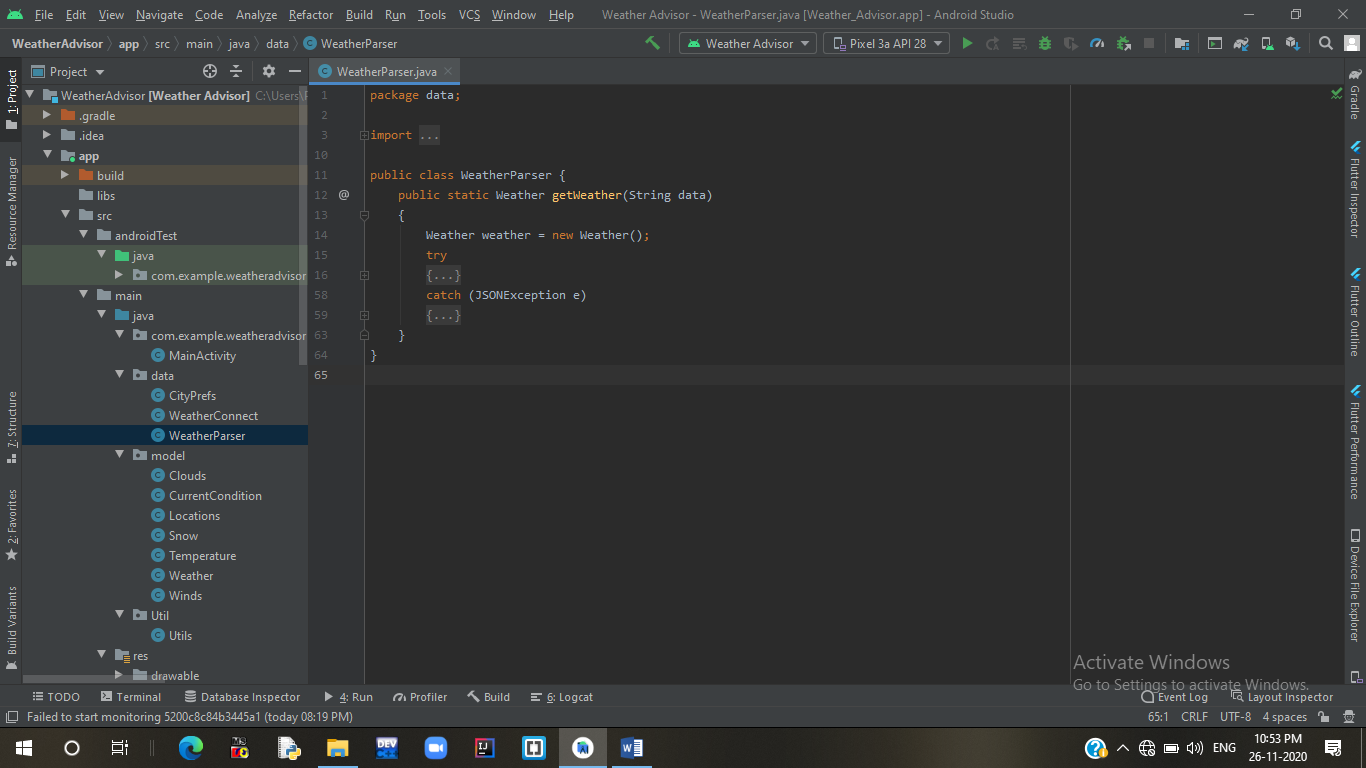




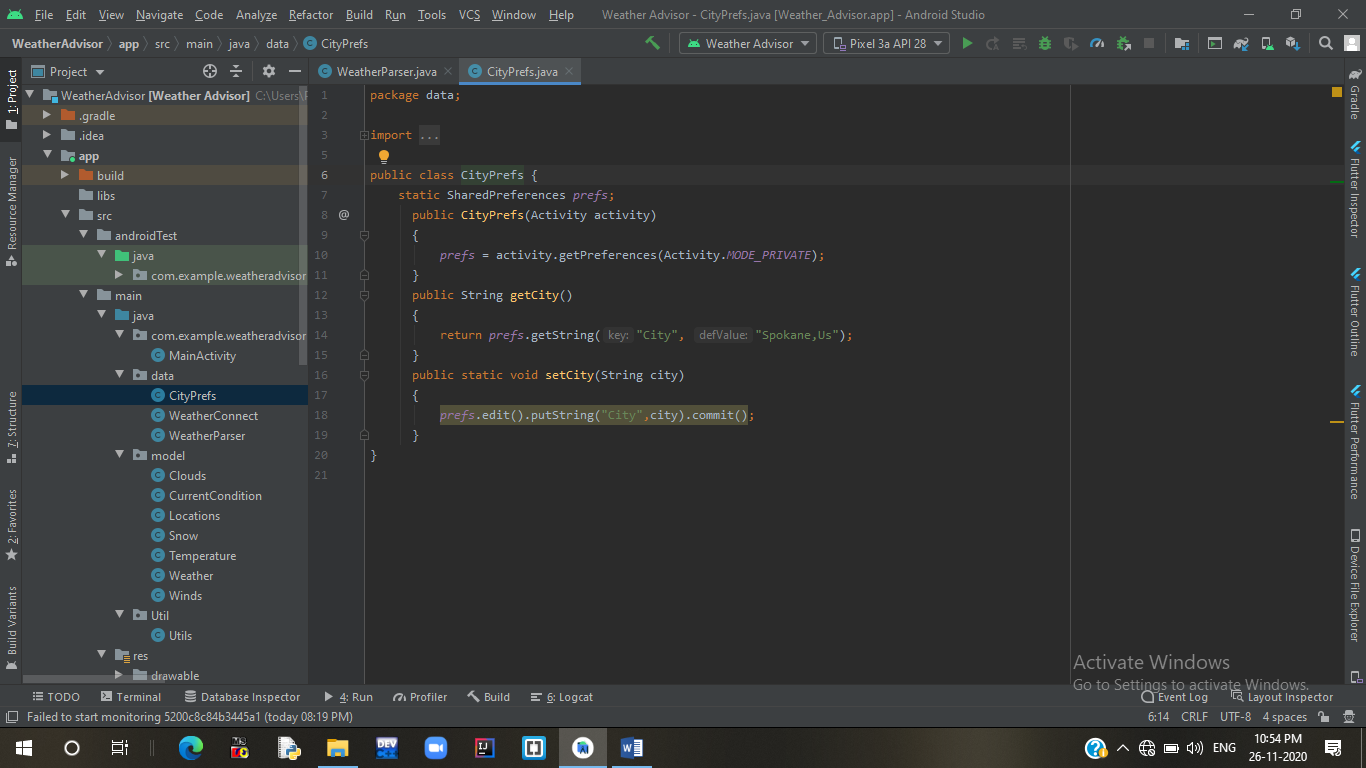
File : WeatherConnect.java



File : WeatherParser.java



File : CityPrefs.java



**References**

* Data from openweathermap.org
* Images from Google
* W3School.com
* Some Youtube Videos