



## ANDROID TEAM - WEEK 3

Hope you all are ready with your responsive UI for the weather app. Now it's time for us to make the application real time. That is we'll integrate the API (Application Programming Interface) to our weather app.

To make the app more responsive, make sure you have the following dependencies in your build.gradle(:app) file:

```
implementation 'androidx.appcompat:appcompat:1.4.1'
implementation 'com.google.android.material:material:1.5.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
implementation 'com.squareup.picasso:picasso:2.71828'
implementation 'com.android.volley:volley:1.2.1'
implementation 'com.google.android.gms:play-services-location:19.0.1'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.3'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
```

**Note:** If any one of the above dependencies is missing in your gradle file add that specific dependency in your gradle file so that you will not get errors when you work with API.



Picasso – <https://github.com/square/picasso>

Volley –

<https://www.geeksforgeeks.org/volley-library-in-android/>

<https://google.github.io/volley/>

In AndroidManifest.xml:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

API integration:

WeatherAPI.com provides access to weather and geo data via a JSON/XML restful API. It allows developers to create desktop, web and mobile applications using this data very easy.

Url for API:

<https://www.weatherapi.com/docs/>

Requirements:



Using mobile GPS fetch the current location which has to be displayed in the textview present at the top.

From the json response of the weather API call for your current location, fetch the following data:

1.temperature

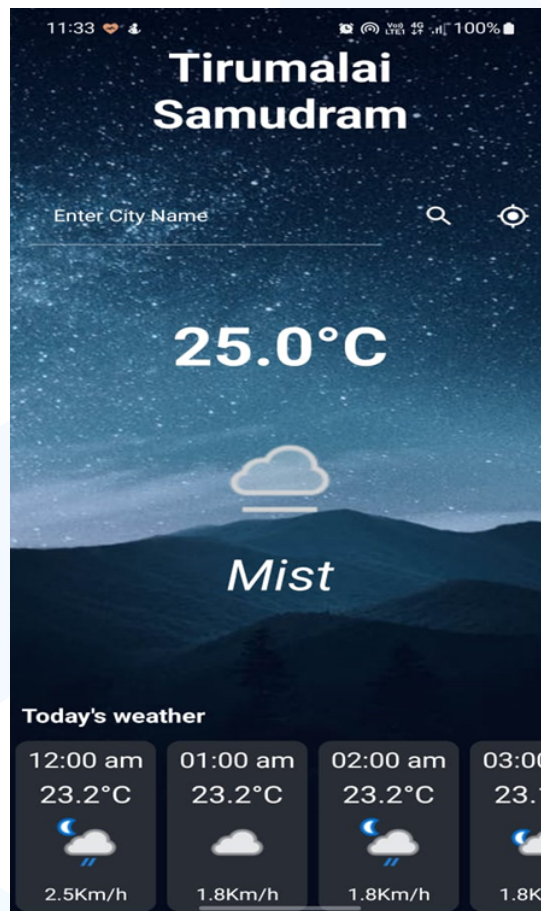
2.condition text

3.condition icon

Reference materials:

<https://www.geeksforgeeks.org/parse-json-java/>

# TECH ODYSSEY



Note: Ignore recycler view for this week's task

## THANK YOU!