Weather App

BY

ARITRA KUNDA(41)
VILAYAT ALI MIRZA(43)
SAPTARSHI MITRA(44)

INDEX

1.SCOPE OF THE PROJECT

Weather is the state of the atmosphere, describing for example the degree to which it is hot or cold, wet or dry, calm or stormy, clear or cloudy. Most weather phenomena occur in the lowest level of the atmosphere, the troposphere, just below the stratosphere. Weather refers to day-to-day temperature and precipitation activity, whereas climate is the term for the averaging of atmospheric conditions over longer periods of time. When used without qualification, "weather" is generally understood to mean the weather of Earth

The scope of this project was to create a weather application on android which would forecast weather based on current location as well as on a 5 day or a 3 day basis. We used openweathermap.org's API to get the current weather as well as 5 day / 3 day weather data.

5 day analysis: In case of 5 day analysis we are simply displaying max and min temperatures of 5 days.

3 day analysis: In case of 3 day analysis we are displaying max and min temperatures as well as humidity and pressure.

Search by city: This displays 5 data analysis based on the city entered.

OpenWeatherMap API: It is one of the most widely used API for weather based services. It is a small IT company established in 2014.There are more than 1,200,000 users and around 1500 users per day.

2.METHODOLOGY FOLLOWED:

This project used Requirements ⟶ Design ⟶ Implementation approach.

Requirements: The requirements of the apps were :

- To display 5-day weather data(with minimised data) on opening the app.
- To display 3-day weather data(with more data such as pressure and temperatures)
- To display weather data based on query of a particular city.

Design: The design of the app consists of a left menu-bar, a top bar with refresh. The menu bar consists of 5-day data,3-day data and search by city.

Implementation:The implementation consisted of mainly getting the weather data using retrofit from Openweathermap.org using the GET request. The API returns the data in the form of JSON and we needed to convert in POJO format. We passed our query in the form of city and openweathermap.org returned the 5 day data in 3 hour format.

3.TECHNOLOGY USED:

API (Application Programming Interface): An API is a set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system. It provides developers with standard commands for performing common operations so they do not have to write the code from scratch.

APIs are available for both desktop and mobile operating systems. The Windows API, for example, provides developers with user interface controls and elements, such as windows, scroll bars, and dialog boxes. It also provides commands for accessing the file system and performing file operations, such as creating and deleting files. Additionally, the Windows API includes networking commands that can be used to send and receive data over a local network or the Internet.

Mobile APIs, such as the iOS API, provide commands for detecting touchscreen input, such as tapping, swiping, and rotating. It also includes common user interface elements, such as a pop-up keyboard, a search bar, and a tab bar, which provides navigation buttons the bottom of the screen. The iOS API also includes predefined functions for interacting with an iOS device's hardware, such as the camera, microphone, or speakers.

Operating system APIs are typically integrated into the software development kit for the corresponding program. For example, Apple's Xcode IDE allows developers to drag and drop elements into an application's interface. It also provides a list of available functions and includes syntax highlighting for known elements and commands.

While operating system APIs have a robust set of features, other types of APIs are much more basic. For example, a website may provide an API for web developers that allows them to access specific information from the site. A website API may be as simple as a set of XML elements with a few basic commands for retrieving the information.

Retrofit: Retrofit is basically a type safe HTTP client for Android and Java. It basically models REST endpoints

as Java interfaces making them simple to work with. It models over base URL and by making interfaces return the entities from a rest endpoint provided the URL returns a JSON.

POJO: (Plain Old Java Object): It is an ordinary Java object, not bound by any special restriction other than those forced by the Java Language Specification and not requiring any class path. POJOs are used for increasing the readability and re-usability of a program.

JSON:(Javascript Object Notation): it is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.

An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An object is an unordered set of name/value pairs. An object begins with {left brace and ends with } right brace. Each name is followed by : colon and the name/value pairs are separated by ,comma.

An array is an ordered collection of values. An array begins with [left bracket and ends with] right bracket. Values are separated by , comma.

A value can be a string in double quotes, or a number,
or true or false or null, or an object or an array. These structures can be nested.

A string is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.

A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

Whitespace can be inserted between any pair of tokens. Excepting a few encoding details, that completely describes the language.

Location Manager API: A data class representing a geographic location.A location can consist of a latitude, longitude, timestamp, and other information such as bearing, altitude and velocity.All locations generated by the LocationManager are guaranteed to have a valid latitude, longitude, and timestamp (both UTC time and elapsed real-time since boot), all other parameters are optional.

GSON(Google Open Source Format): Google Gson is an open source, Java-based library developed by Google. It facilitates serialization of Java objects to JSON and vice versa. This tutorial adopts a simple and intuitive way to describe the basic-to-advanced concepts of Google Gson and how to use its APIs. It is used to serialize /desrialise objects in JSON.

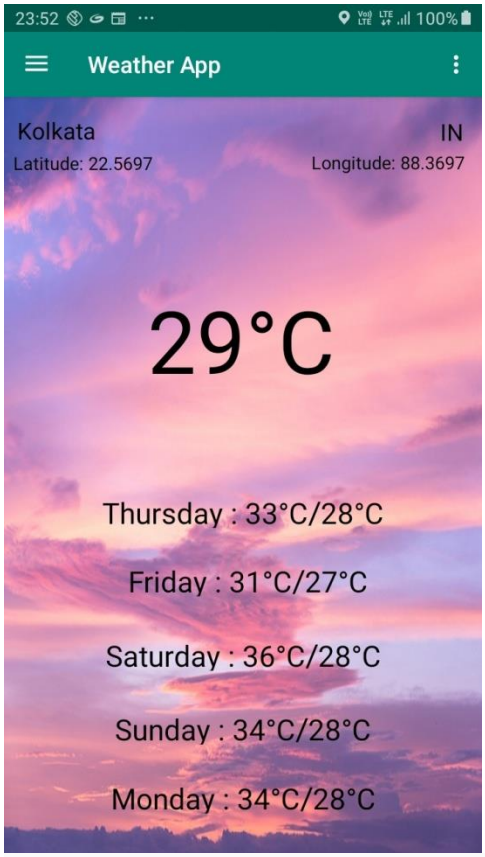
4.IMPLEMENTATION:

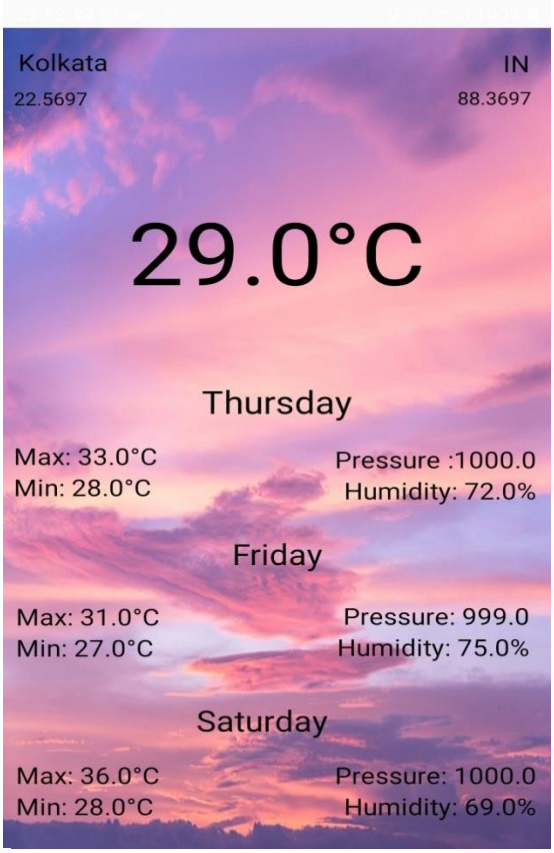
The implementation of this project went through the following procedures:

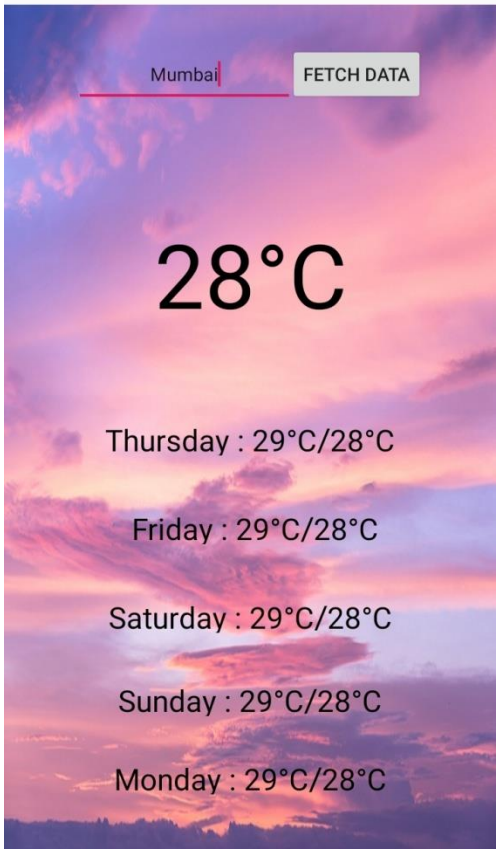
1. Using get request to fetch the weather data from openweathermap.org: Openweathermap API returns weather data in the form of JSON. Then we used POJO class to get that data and returned that data in the respective fields.

2. Using 3 different menus for 5,3 days data and search by city: When the user opens the app he/she sees 5 day data and a menu bar on the left. On expanding it there are 3 items in the menu,5-day forecast,3-day forecast and search by city. When clicking 5-day data he/she sees max and min temperatures of 5 days. When clicking 3-day data he/she sees max,min temperatures as well as pressure and humidity. When clicking search by city data he/she gets weather data of a particular city in format of 5 days,that is max/min temperatures of 5 days. We are achieving this whole concept by creating new Intent on selecting these items, and using suitable methods for them.

3. Refresh:There is a 3 dots button in the top bar which refreshes the current data. We are basically calling the same method for displaying the current data.


23:52
100%
Weather App
Kolkata
IN
Latitude: 22.5697
Longitude: 88.3697
29°C
Thursday : 33°C/28°C
Friday : 31°C/27°C
Saturday : 36°C/28°C
Sunday : 34°C/28°C
Monday : 34°C/28°C


Kolkata
IN
22.5697
88.3697
29.0°C
Thursday
Max: 33.0°C
Min: 28.0°C
Pressure :1000.0
Humidity: 72.0%
Friday
Max: 31.0°C
Min: 27.0°C
Pressure: 999.0
Humidity: 75.0%
Saturday
Max: 36.0°C
Min: 28.0°C
Pressure: 1000.0
Humidity: 69.0%


Mumbai
FETCH DATA
28°C
Thursday : 29°C/28°C
Friday : 29°C/28°C
Saturday : 29°C/28°C
Sunday : 29°C/28°C
Monday : 29°C/28°C

5.CONCLUSION/FUTURE PROSPECTS

This project has a lot of scope of improvement.

Optimisation: The app can be made more optimized by developing a better logic and utilization of resources efficiently. There are several places where we can improve the logic of the code and optimize the app.

Making the app more visual: The app can be made more appealing to the user by usage of icon packs for temperatures as well as humidity and pressure.There is a dedicated method in Openweathermaps API which returns icon based on the weather location.We could use that method in our app.