

DÉPARTEMENT INFORMATIQUE APPLIQUÉE

FACULTÉ DE TECHNOLOGIE DE L'INFORMATION ET DE LA COMMUNICATION

**Programmation mobile hybride**

Flutter Project Reports – Weather App

Professor:

* Shiam Beehary.

Done by:

* RAKOTONDRAMANANA ANDRIAMANANJARA AMBOARAPITIA LAFATRA.
* MAHEFAHERIVOLA Feelminho.

Table of content:

1. Description of the project. ................................................................. 03
2. Objective. ..................................................................................... 03
3. Key features. ................................................................................. 03
4. Technologies used. ....................................................................... 04
5. Key Features of Weather APIs. ...................................................... 04
6. User interface/User Experience design. ............................................. 06
7. Mockup. ........................................................................................ 06
8. Wireflow. ...................................................................................... 06
9. Explanation of the activities used in the application. ......................... 07
10. Home page. .................................................................................. 07
11. Hourly forecast. ............................................................................ 08
12. City search. ................................................................................... 09
13. Weather forecast page. ................................................................ 10
14. Explanation of Dart main section of codes. ........................................ 11
15. Possible improvements. .................................................................... 12
16. Conclusion. ........................................................................................ 13
17. **Description of the project**

Our project involves creating a mobile application using Flutter and Dart. The application will focus on the theme of weather and will be named "Weather - App".

1. Objective:

The main objective of this application is to provide users with intuitive and easy-to-understand weather information. By centralizing the most recent weather data, "Weather - App" will offer a user-friendly and convenient experience for users looking to plan their activities based on weather conditions.

1. Key Features:
   1. Display of Current Data Users will be able to see real-time weather conditions for the location they are interested in. This will include temperature, humidity, wind speed, as well as other relevant information.
   2. Short and Long-Term Forecasts The application will provide weather forecasts for the upcoming hours of the day. Users will have an overview of the changes to expect in weather conditions.
   3. Weather Search for a City For practical and personalized use, users can search for a city in a search bar to view the weather forecast for that specific location.
   4. User-Friendly Interface The application will be designed with an intuitive and user-friendly interface, allowing for easy navigation and an enjoyable experience for all users.
2. Technologies Used:

3.1 Flutter & Dart We will use the Flutter framework, based on the Dart programming language, for the development of the application. Flutter offers high performance, adaptable design, and ease of development, making it an ideal choice for our project.

3.2 Weather APIs Weather APIs (Application Programming Interfaces) are a set of protocols and tools for extracting weather data from various sources. Acting as an intermediary, they allow software applications to access real-time forecasts and historical weather data. These APIs aggregate information from weather stations, satellites, radars, and other sources, providing structured and easily exploitable weather details.

4. Key Features of Weather APIs:

* 1. Real-Time Access:

Weather APIs offer the ability to access crucial data in real-time such as current temperature, humidity, wind speed and direction, precipitation, and other relevant weather conditions.

* 1. Historical Weather Data:

They also provide access to historical weather data, essential for research, analysis, and planning.

* 1. Weather Forecasts:

APIs provide short and long-term forecasts, essential for planning weather-sensitive activities.

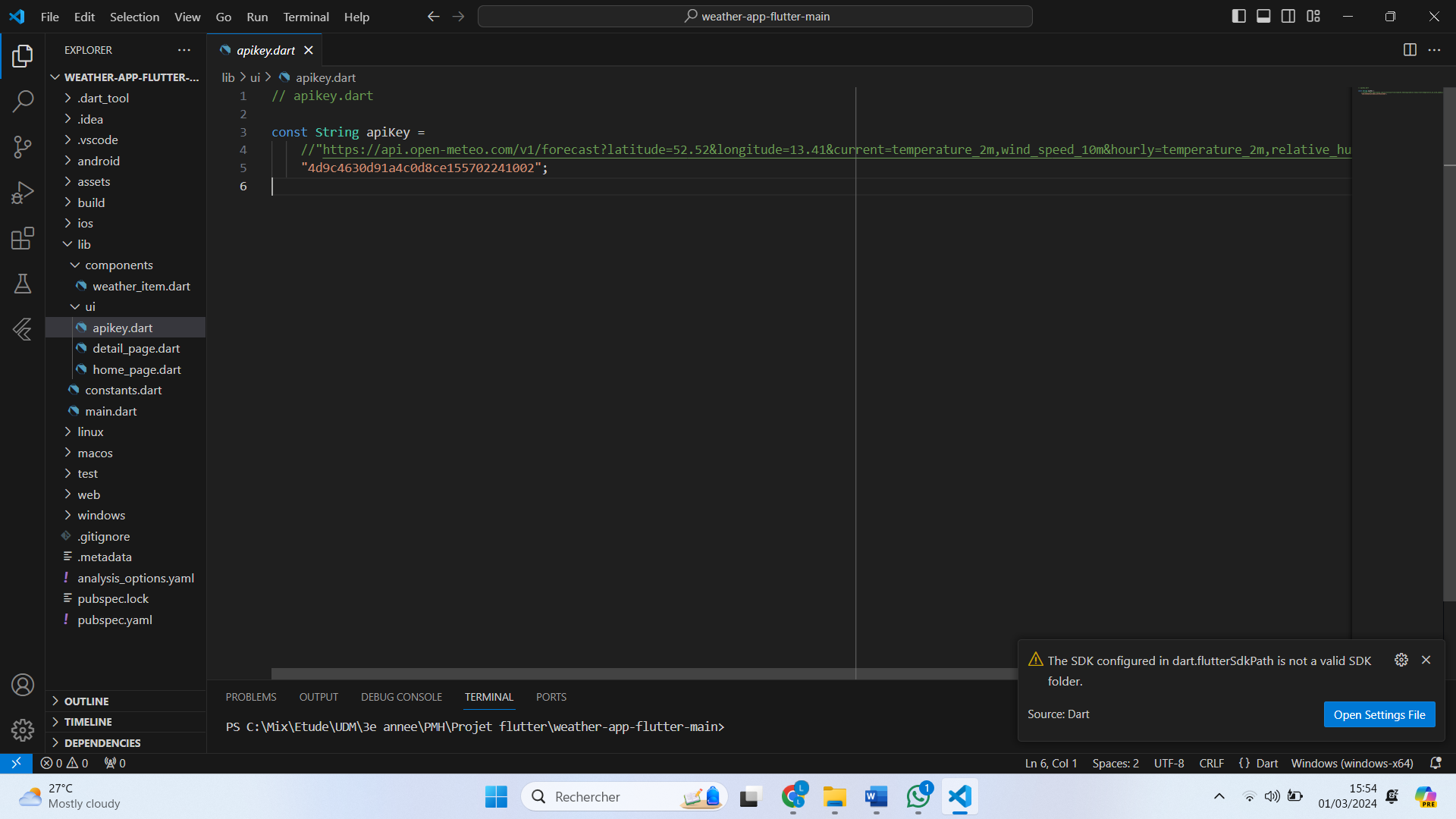
* 1. Global Coverage:

Many weather APIs offer weather data on a global scale, facilitating international operations.

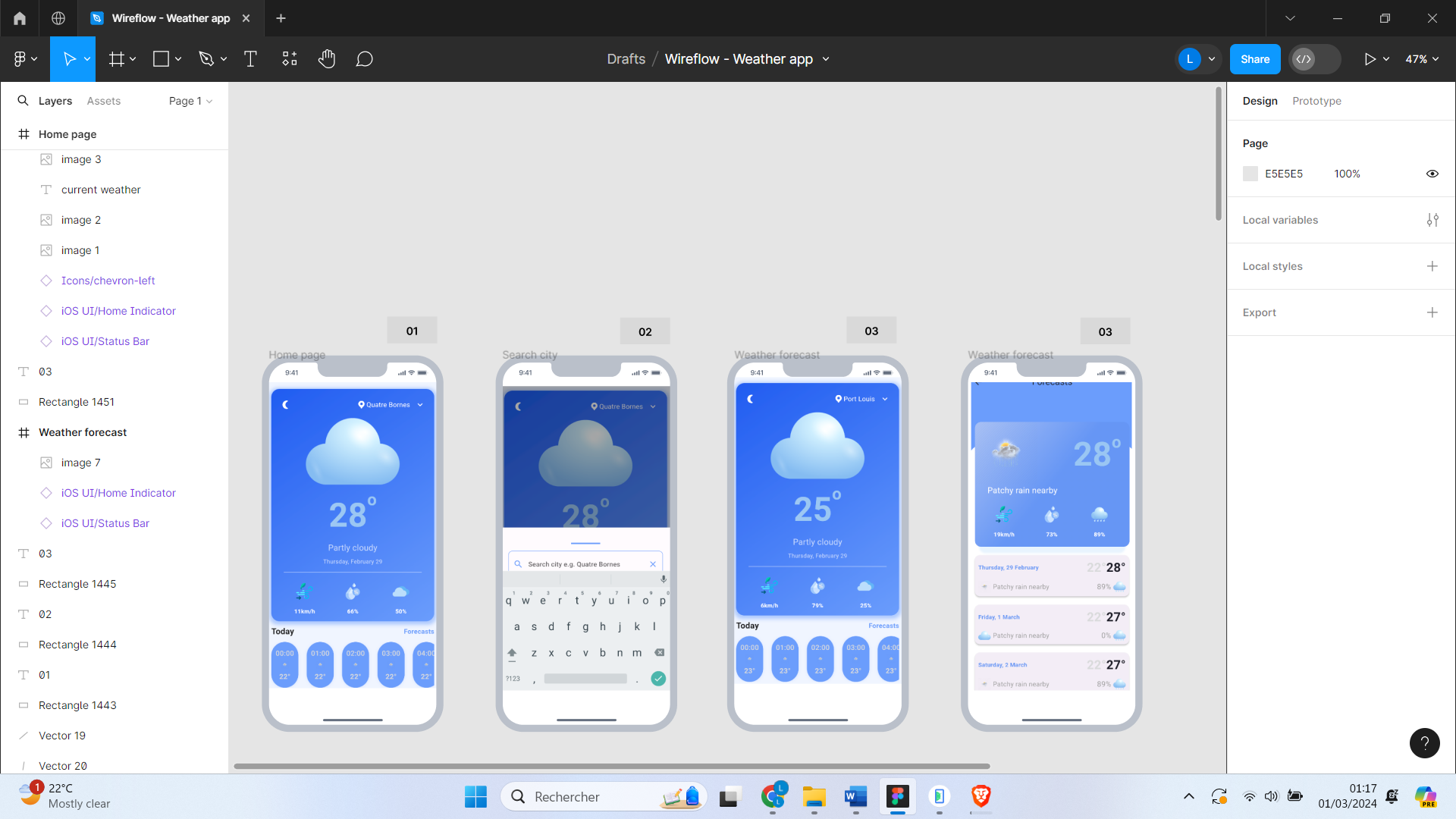
* 1. Customization:

Depending on the API, users can specify types of weather data, locations, and specific times, ensuring great versatility.

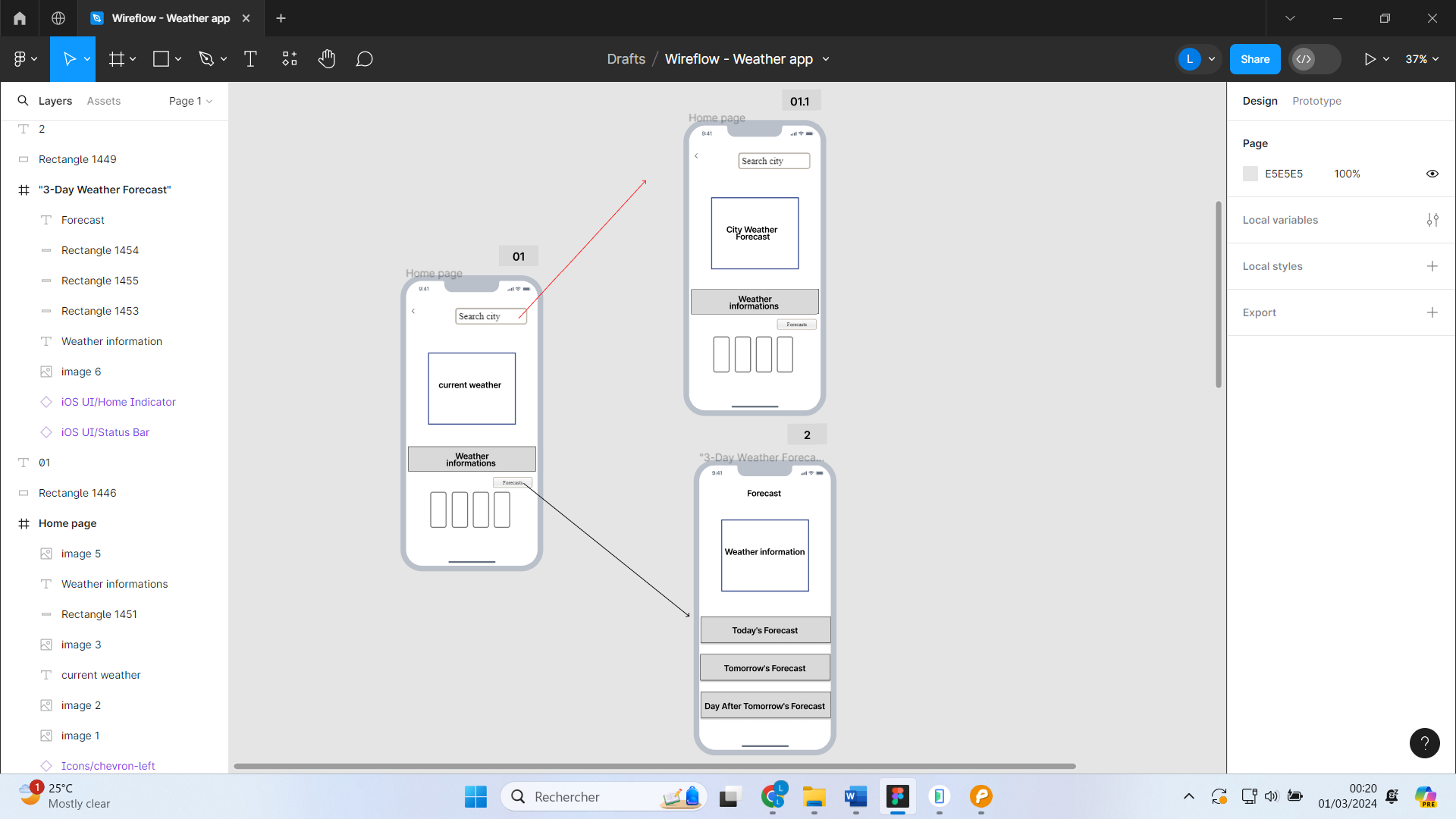
To obtain real-time weather data and forecasts, we will integrate reliable and accurate weather APIs from the website <https://www.weatherapi.com/>. This will ensure up-to-date and quality information for users of "Weather - App".



1. **User interface/ User Experience design.**
2. Mockup:

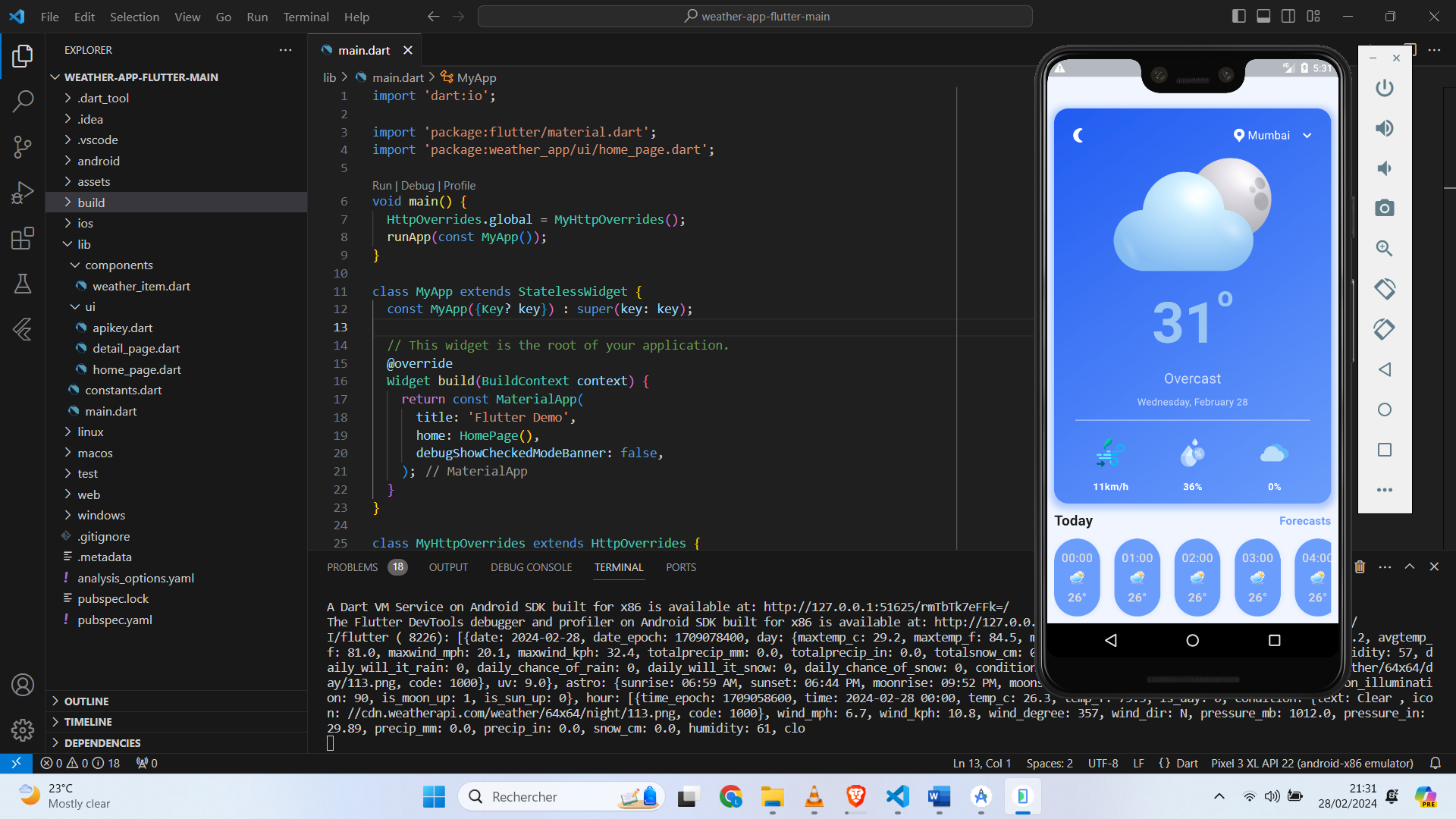


1. Wireflow:



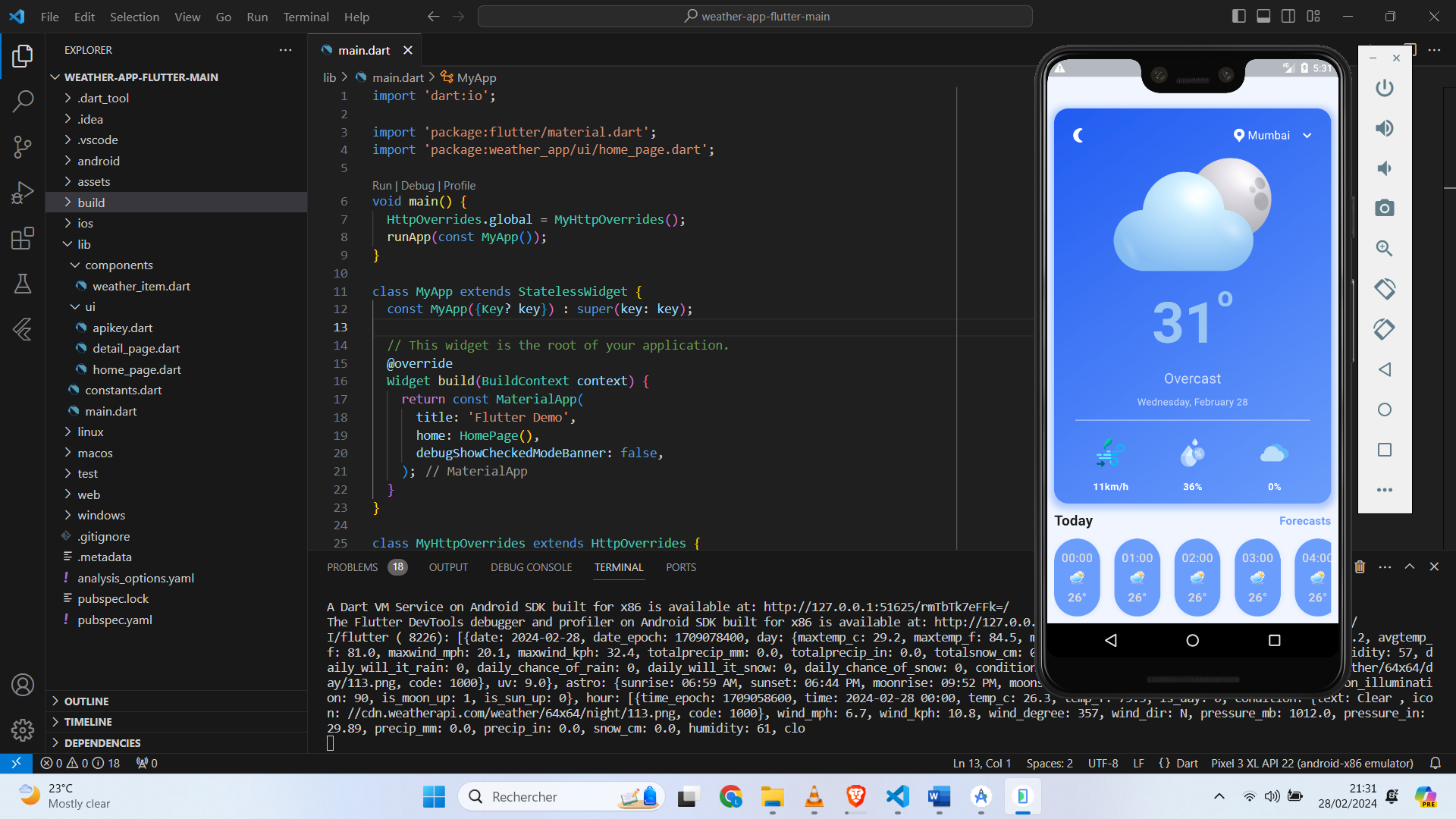
1. **Explanation of the Activities used in the application with screenshots and short description.**
2. Home Page:

The home page displays the current weather information of the user's default location. This includes the temperature, humidity, wind speed, and a brief description of the current conditions.



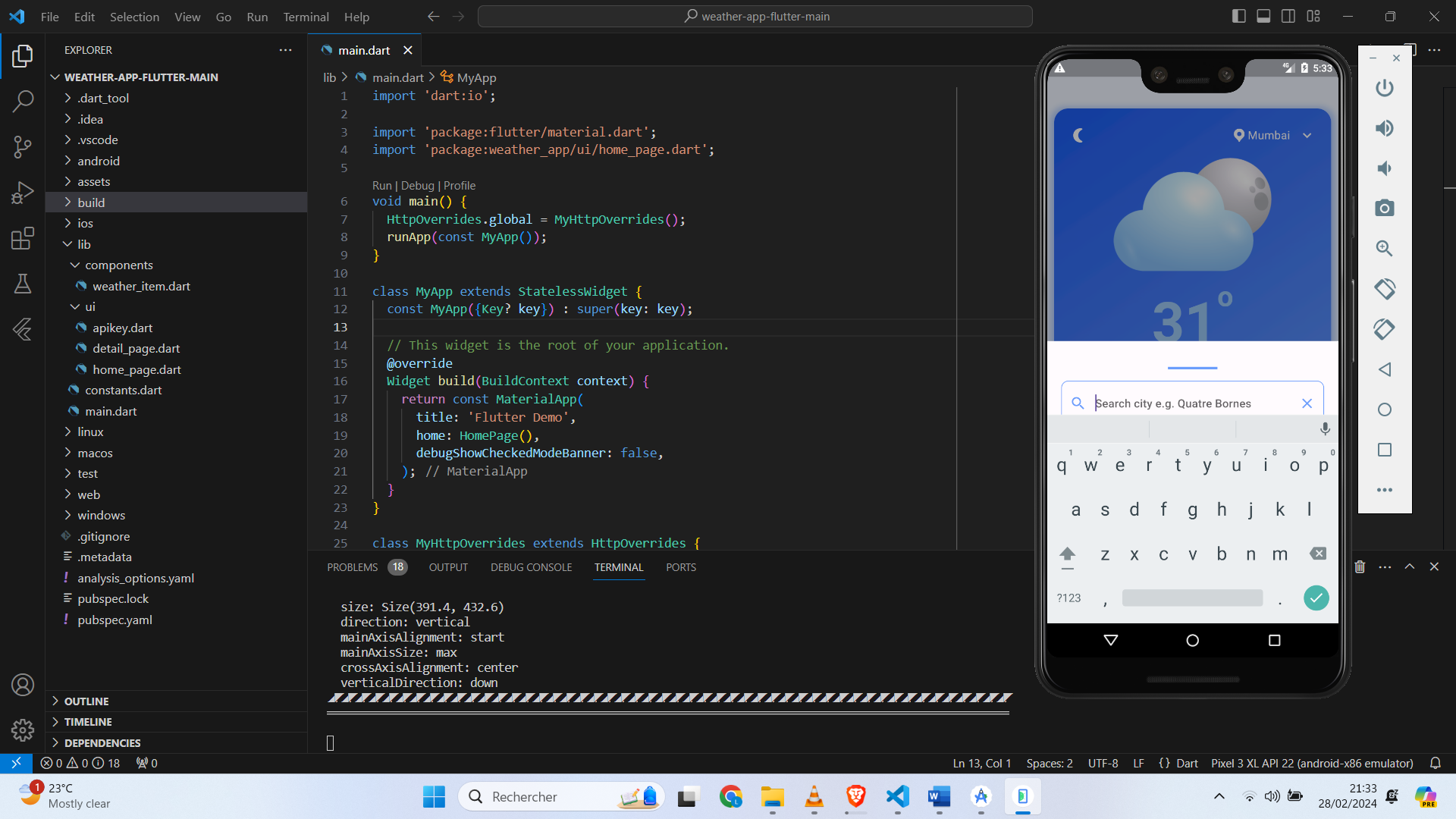
1. Hourly Forecast:

A dedicated section for hourly forecasts allows users to view the predicted weather changes for the upcoming hours. Clear and informative graphs illustrate the trend of temperatures and precipitation at the bottom of the application interface.



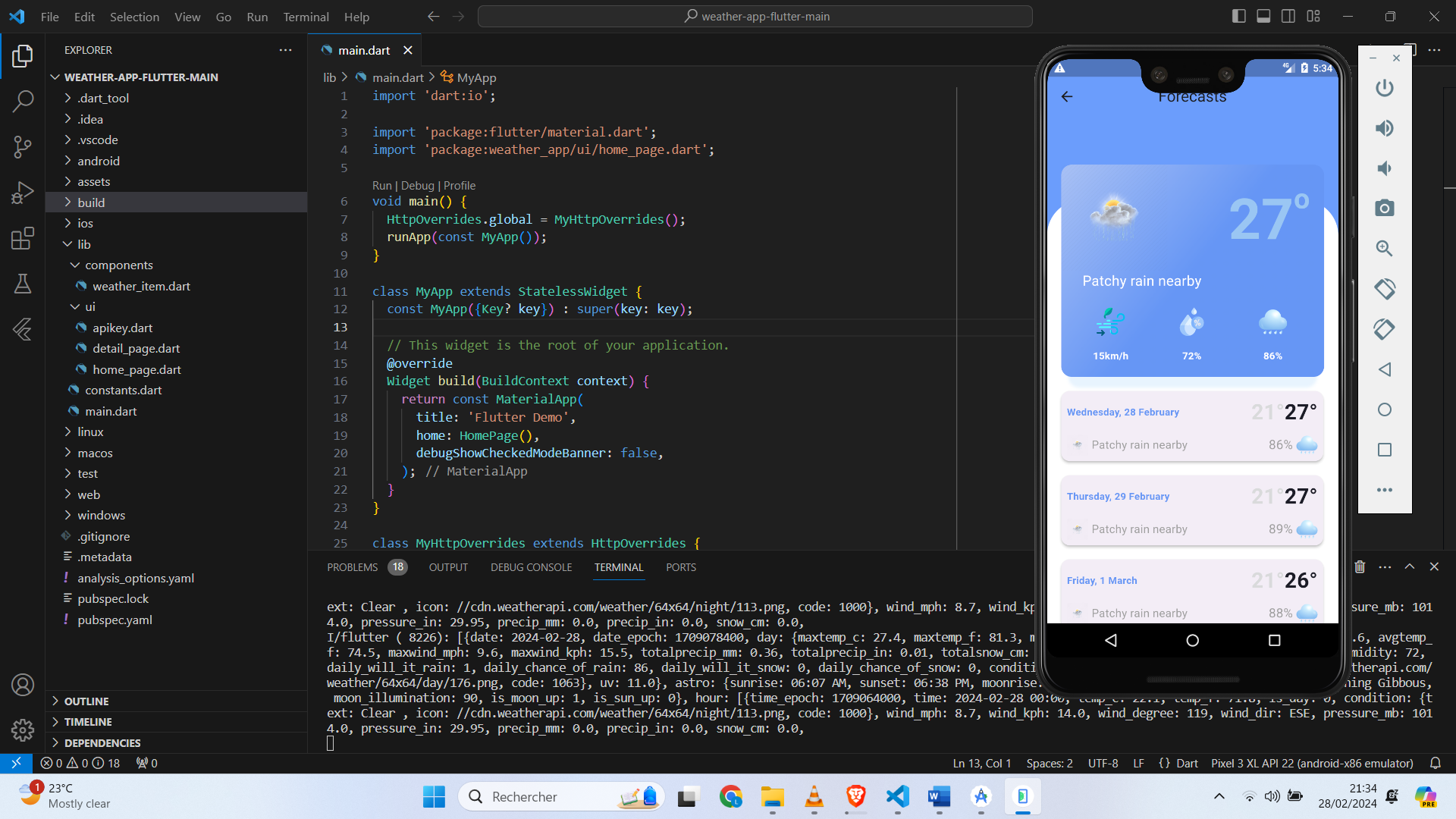
1. City Search:

Using the search bar, users can enter the name of a city to get specific weather forecasts for that location. The results include the temperature, current weather conditions, as well as short-term forecasts.



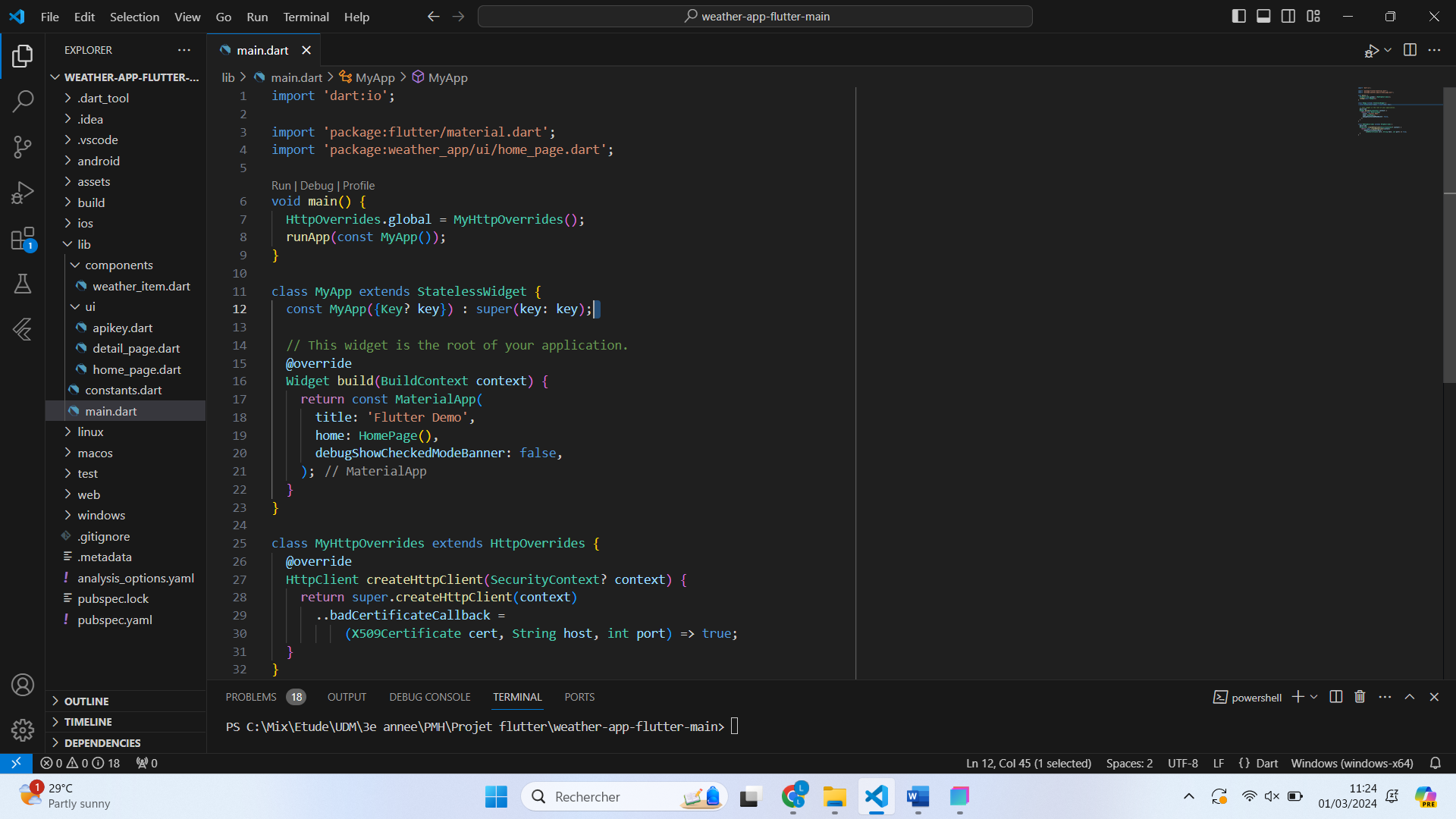
1. Weather Forecast Page:

When you click on the "Forecast" button, it takes you to a dedicated page displaying in detail the weather forecasts for the next two days. On this page, you will find information such as the predicted maximum and minimum temperatures, the forecasted weather conditions (such as sunny, cloudy, rainy, etc.), as well as other relevant data like chances of precipitation, expected winds, and perhaps even weather tips to plan your activities accordingly.



1. **Explanation of Dart main section of codes.**

The Main.dart code is the starting point of the Flutter application to display the weather app. It imports necessary libraries, defines a main class MyApp that extends StatelessWidget, and overrides the global configuration for HTTP requests to ignore invalid SSL certificates. The build method of MyApp constructs the user interface using the MaterialApp widget, which specifies the app title, the home page, and disables the debug banner. The MyHttpOverrides class extends HttpOverrides and defines a callback function to ignore invalid SSL certificates during HTTP requests. This code configures the Weather-App to display a weather user interface, customizing the behavior of HTTP requests to handle invalid SSL certificates.



1. **Possible improvements.**

Despite the app's strong features, there are areas where improvements can be made to provide an even better experience for users:

1. Adding Notifications

Integrate the ability for users to receive notifications for important weather alerts such as storms, heavy showers, or sudden temperature changes.

1. Customization of Preferences

Offer users the option to customize their weather preferences, such as units of measurement (Celsius vs Fahrenheit), priority data types to display, etc.

1. Improving City Search

Refine the search algorithm for better accuracy when searching for cities, including automatic suggestions based on geolocation.

1. Adding Social Features

Integrate social features such as sharing current weather conditions on social media, or the ability to create groups to share specific weather forecasts.

1. Performance Optimization

Continue to optimize the app's performance to ensure a smooth and responsive experience, even on older or less powerful mobile devices.

By considering these suggestions for improvement, "Weather - App" can evolve to become a comprehensive weather application highly appreciated by its users, offering not only accurate and updated information but also a personalized and social weather experience.

1. **Conclusion**

In conclusion, the "Weather - App" project aims to provide users with a friendly and convenient weather experience using Flutter and Dart for app development. With features such as displaying current data, short and long-term forecasts, city weather search, and a user-friendly interface, the app offers quick and easy access to the most relevant and up-to-date weather information.

The screenshots above clearly illustrate the various activities of the app, showing how users can obtain detailed hourly forecasts, search for weather forecasts for specific cities, and plan accordingly with forecasts for the next two days.