Smart Weather Forecast Application using Flutter

Submitted by: Riya Parikh (22MID0356)

Abstract

This document provides a comprehensive overview of the **Smart Weather Forecast Application**, a Flutter-based project designed to display real-time weather data using a public API. The app integrates **HTTP requests**, **Lottie animations**, and a **responsive UI** to enhance user experience. It also demonstrates fundamental concepts of Flutter such as widget structuring, state management, and asynchronous programming.

Table of Contents

1. Introduction

The Smart Weather Forecast App provides users with accurate and up-to-date weather information for any city they enter. Unlike conventional weather apps, it uses a free and open JSON endpoint (wttr.in) that requires no authentication. Built using Flutter, it runs seamlessly on Android, iOS, and Web platforms.

The project aims to combine beautiful UI design with practical functionality, creating a well-rounded academic submission that reflects modern mobile app development practices.

2. Objectives

- Develop a cross-platform weather application using Flutter.

- Fetch and parse weather data in real time using HTTP requests.
- Display weather parameters (temperature, humidity, conditions) with engaging animations.
- Maintain clean, modular, and well-documented code.
- Avoid external sign-ups or authentication barriers for simplicity.

3. Tools & Technologies

Tool / Library	Purpose
Flutter SDK	Framework for building the UI and logic
Dart	Primary programming language
HTTP Package	Fetches API data from wttr.in
Lottie Package	Displays animated weather icons
VS Code	IDE used for coding and debugging

4. System Design and Project Structure

The architecture follows a simple three-layer structure:

1. UI Layer – Built using Flutter widgets such as `Scaffold`, `TextField`, and `Card`.
2. Logic Layer – Handles user input, API requests, and error handling.
3. Presentation Layer – Displays the data dynamically with animations and smooth transitions.

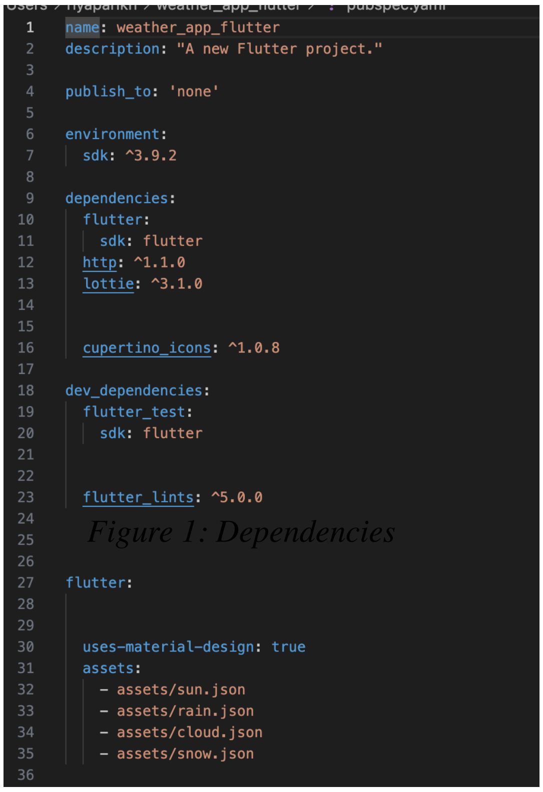
The data flow can be summarized as:

User Input → HTTP Request → JSON Response → Parsed Data → UI Update5. Project Structure

```
weather_app_flutter/
│
├── assets/                ← all Lottie animation files
│   ├── sun.json
│   ├── rain.json
│   ├── cloud.json
│   └── snow.json
│
├── lib/
│   └── main.dart          ← main Flutter code (UI + logic)
│
├── test/
│   └── widget_test.dart     ← (default test file)
```

```
|
├──── pubspec.yaml            ← contains dependencies + assets paths
├──── analysis_options.yaml
├──── README.md
├──── .gitignore
├──── .metadata
└──── .idea/ or .vscode/      ← IDE configs (optional, depends on your editor)
```

6. Implementation & Code Explanation

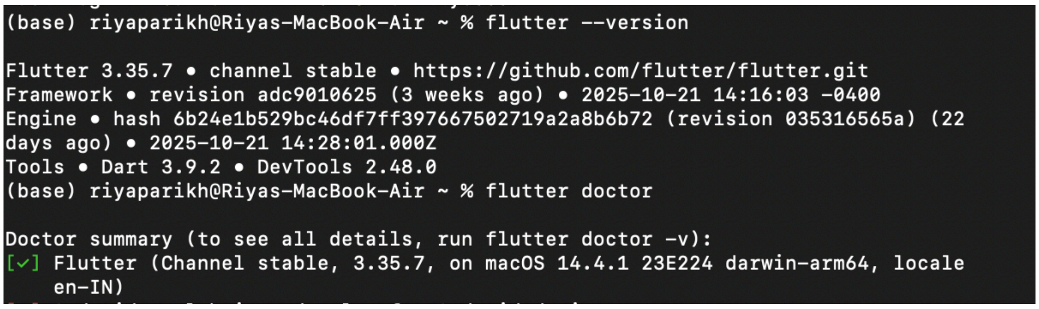


```yaml
name: weather_app_flutter
description: "A new Flutter project."

publish_to: 'none'

environment:
  sdk: ^3.9.2

dependencies:
  flutter:
    sdk: flutter
  http: ^1.1.0
  lottie: ^3.1.0


  cupertino_icons: ^1.0.8

dev_dependencies:
  flutter_test:
    sdk: flutter


  flutter_lints: ^5.0.0


flutter:


  uses-material-design: true
  assets:
    - assets/sun.json
    - assets/rain.json
    - assets/cloud.json
    - assets/snow.json
```

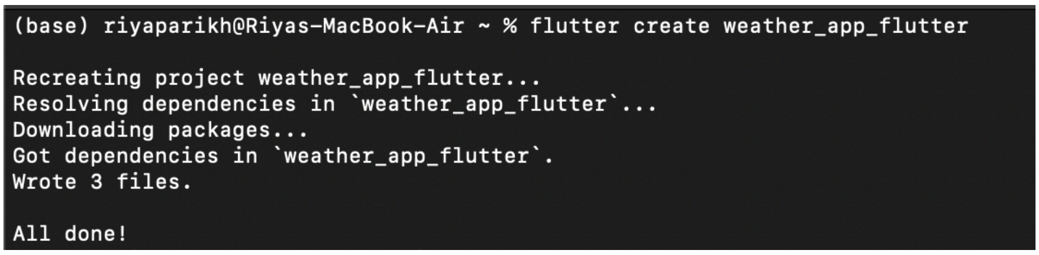
Figure 1: Dependencies

```
import 'package:flutter/material.dart';

border: OutlineInputBorder(
borderRadius: BorderRadius.circular(15),
borderSide: BorderSide.none,),),),
const SizedBox(height: 20),
ElevatedButton(
onPressed: () {
final city = _cityController.text.trim();
if (city.isNotEmpty) fetchWeather(city);
},
child: const Text('Get Weather'),
),
const SizedBox(height: 30),
if (_loading)
const CircularProgressIndicator()
else if (_weatherData != null)
Expanded(child: buildWeatherCard())
else
const Text('Enter a city to view weather ☁️'),
],),),);}

Widget buildWeatherCard() {
final current = _weatherData!['current_condition'][0];
final desc = current['weatherDesc'][0]['value'];

return Card(
shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),
elevation: 8,
color: Colors.white,
child: Padding(
padding: const EdgeInsets.all(20.0),
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Lottie.asset(getWeatherAnimation(desc), height: 150),
const SizedBox(height: 15),
Text(desc, style: const TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),
const SizedBox(height: 10),
Text('Temperature: ${current['temp_C']}°C', style: const TextStyle(fontSize: 18)),
Text('Humidity: ${current['humidity']}%', style: const TextStyle(fontSize: 18)),
Text('Feels Like: ${current['FeelsLikeC']}°C', style: const TextStyle(fontSize: 18)),
],),),);}}
```

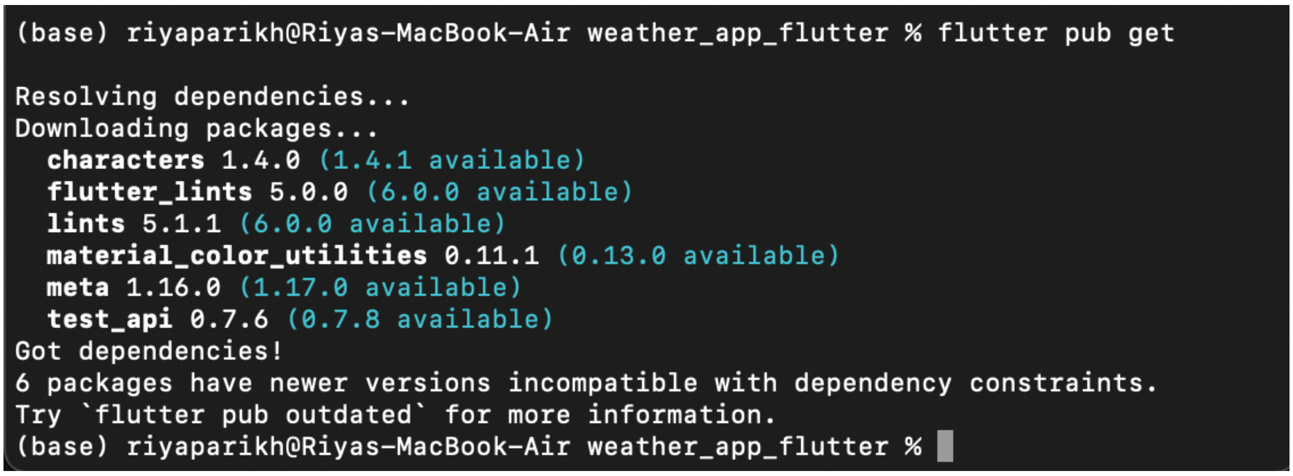
7. Screenshots-Terminal



```
(base) riyaparikh@Riyas-MacBook-Air ~ % flutter --version

Flutter 3.35.7 • channel stable • https://github.com/flutter/flutter.git
Framework • revision adc9010625 (3 weeks ago) • 2025-10-21 14:16:03 -0400
Engine • hash 6b24e1b529bc46df7ff397667502719a2a8b6b72 (revision 035316565a) (22
days ago) • 2025-10-21 14:28:01.000Z
Tools • Dart 3.9.2 • DevTools 2.48.0
(base) riyaparikh@Riyas-MacBook-Air ~ % flutter doctor

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.35.7, on macOS 14.4.1 23E224 darwin-arm64, locale
    en-IN)
```

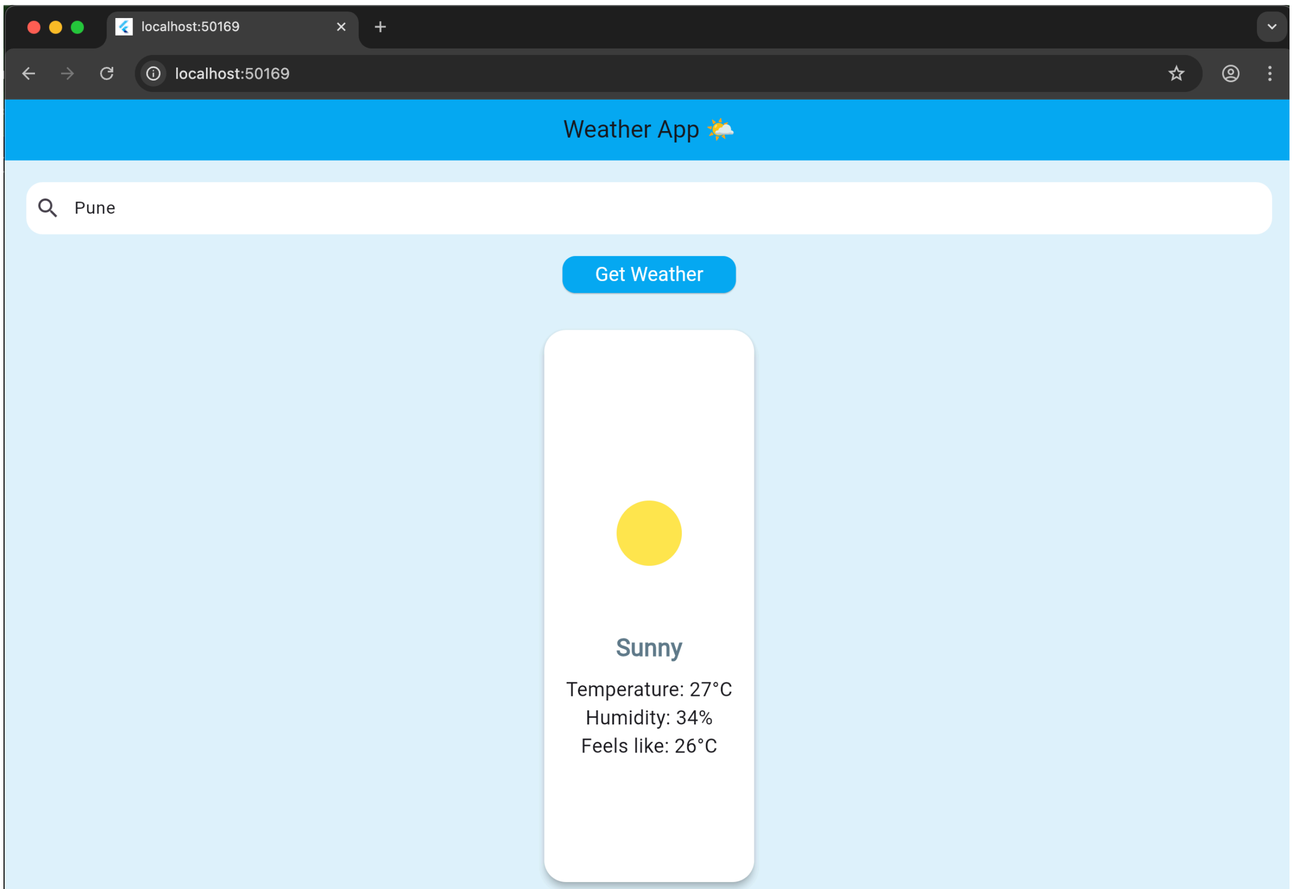


```
(base) riyaparikh@Riyas-MacBook-Air ~ % flutter create weather_app_flutter

Recreating project weather_app_flutter...
Resolving dependencies in `weather_app_flutter`...
Downloading packages...
Got dependencies in `weather_app_flutter`.
Wrote 3 files.

All done!
```

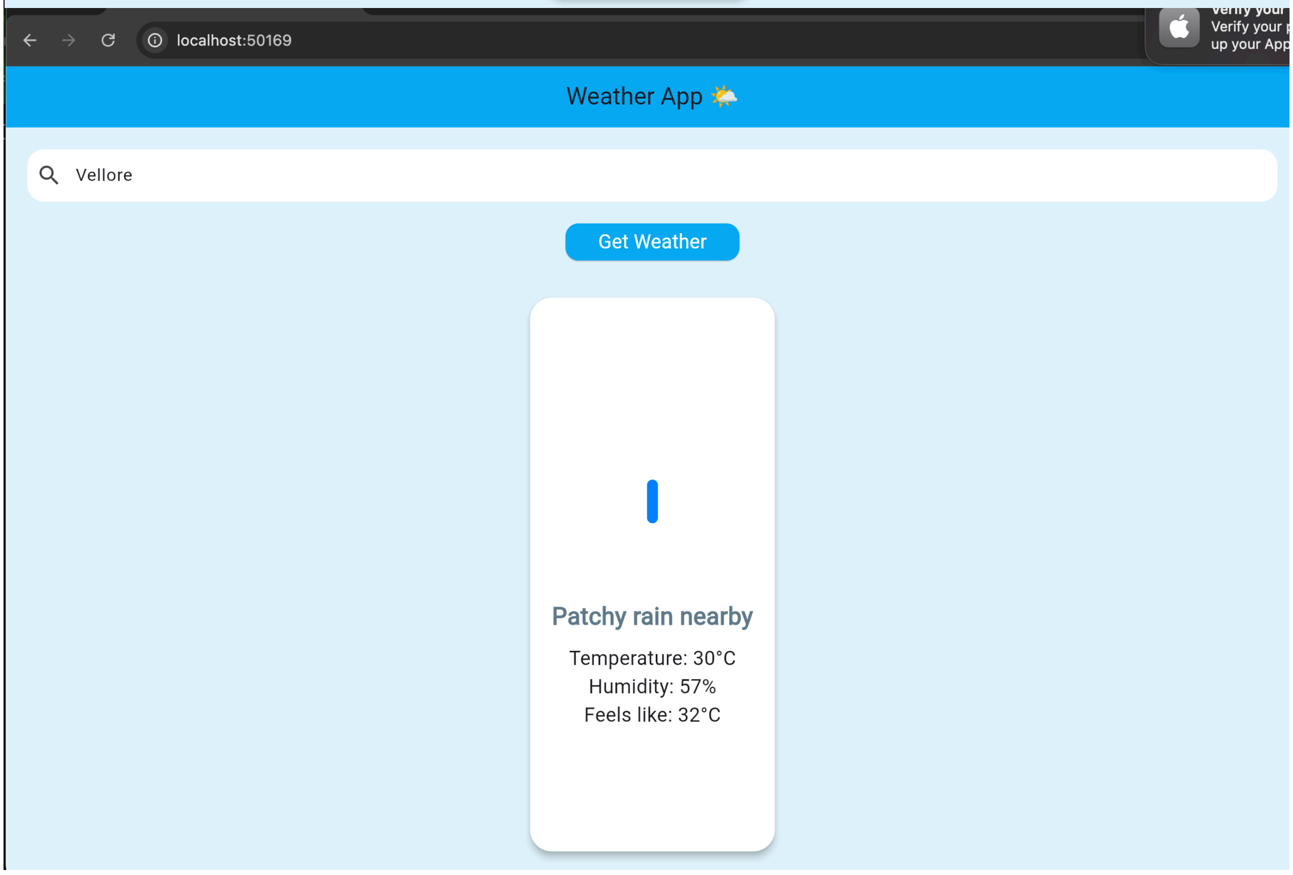


```
(base) riyaparikh@Riyas-MacBook-Air weather_app_flutter % flutter pub get

Resolving dependencies...
Downloading packages...
  characters 1.4.0 (1.4.1 available)
  flutter_lints 5.0.0 (6.0.0 available)
  lints 5.1.1 (6.0.0 available)
  material_color_utilities 0.11.1 (0.13.0 available)
  meta 1.16.0 (1.17.0 available)
  test_api 0.7.6 (0.7.8 available)
Got dependencies!
6 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
(base) riyaparikh@Riyas-MacBook-Air weather_app_flutter %
```

Output:



Sunny
Temperature: 27°C
Humidity: 34%
Feels like: 26°C



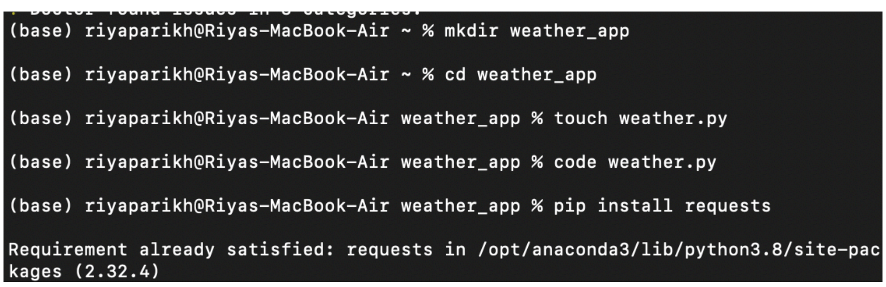
Patchy rain nearby
Temperature: 30°C
Humidity: 57%
Feels like: 32°C

7. Conclusion & Future Scope

The project demonstrates a clean and practical Flutter implementation ideal for academic submission.
Future Enhancements:

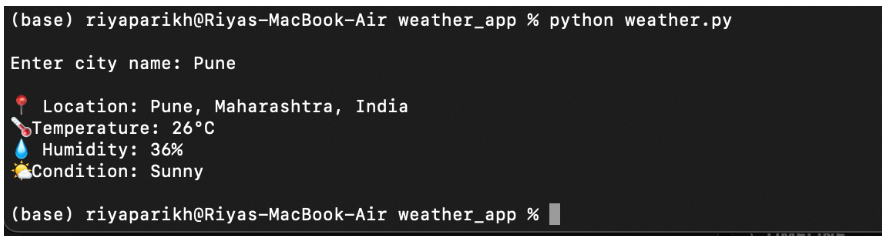
- Add weekly forecast and dynamic backgrounds.
- Integrate user’s GPS location.
- Improve error handling and add dark mode.

WEATHER APP WITHOUT USING FLUTTER



```
(base) riyaparikh@Riyas-MacBook-Air ~ % mkdir weather_app

(base) riyaparikh@Riyas-MacBook-Air ~ % cd weather_app

(base) riyaparikh@Riyas-MacBook-Air weather_app % touch weather.py

(base) riyaparikh@Riyas-MacBook-Air weather_app % code weather.py

(base) riyaparikh@Riyas-MacBook-Air weather_app % pip install requests

Requirement already satisfied: requests in /opt/anaconda3/lib/python3.8/site-pac
kages (2.32.4)
```

```python
import requests
def get_weather(city):
    try:
        # This is a free endpoint (no login or API key needed)
        url = f"https://wttr.in/{city}?format=j1"
        response = requests.get(url)
        data = response.json()

        # Extract important details
        area = data['nearest_area'][0]['areaName'][0]['value']
        region = data['nearest_area'][0]['region'][0]['value']
        country = data['nearest_area'][0]['country'][0]['value']
        temp_c = data['current_condition'][0]['temp_C']
        weather_desc = data['current_condition'][0]['weatherDesc'][0]['value']
        humidity = data['current_condition'][0]['humidity']

        # Print results
        print(f"\n📍 Location: {area}, {region}, {country}")
        print(f"🌡️ Temperature: {temp_c}°C")
        print(f"💧 Humidity: {humidity}%")
        print(f"🌤️ Condition: {weather_desc}\n")

    except Exception as e:
        print("⚠️ Could not fetch weather data. Please check your internet or city name.")
        print("Error:", e)

if __name__ == "__main__":
    city = input("Enter city name: ")
    get_weather(city)
```



```
(base) riyaparikh@Riyas-MacBook-Air weather_app % python weather.py

Enter city name: Pune

📍 Location: Pune, Maharashtra, India
🌡️Temperature: 26°C
💧 Humidity: 36%
🌤️Condition: Sunny

(base) riyaparikh@Riyas-MacBook-Air weather_app % 
```