AIM:

- The primary aim of a live weather app is to provide users with real-time access to accurate and up-to-date weather information for their current location and any other locations they choose. This information should be presented in a clear, concise, and visually appealing way that is easy for users to understand and use.

- The weather application will provide users with real-time weather information, forecasts, and other weather-related data, which can help them make better decisions about their day-to-day activities.

OBJECTIVE:

- Develop a user-friendly Python application that displays real-time weather information for any location.
- Real-time weather data acquisition: Fetch current weather information for specified locations using a reliable weather API.
- User-friendly interface: Develop an intuitive and visually appealing GUI using a Python library like Tkinter or PyQt.
- Location-based weather updates: Allow users to enter city names or use device geolocation to retrieve weather data for their current location.
- Detailed weather information: Display key weather data points like temperature, humidity, wind, precipitation, visibility, UV index, and sunrise/sunset times.
- Multi-location support: Enable saving and switching between multiple locations for easy access.
- Weather alerts: Integrate basic alerts for severe weather conditions.

SOURCE CODE:

```python
import requests

def get_weather(api_key, city_name):
    base_url = "http://api.openweathermap.org/data/2.5/weather"
    params = {"q": city_name, "appid": api_key, "units": "metric"}  # You can change the units if you prefer imperial

    try:
        response = requests.get(base_url, params=params)
        data = response.json()

        if response.status_code == 200:
            temperature = data["main"]["temp"]
            description = data["weather"][0]["description"]
            return f"Temperature: {temperature}°C\nDescription: {description.capitalize()}"
        else:
            return f"Error: {data['message']}"

    except Exception as e:
        return f"Error: {str(e)}"

# Replace 'YOUR_API_KEY' with the actual API key you obtained from OpenWeatherMap
api_key = "73851b2e93b71bbb572a8a704859e96b"
city_name = "Chennai, IN"  # e.g., "London, UK"

result = get_weather(api_key, city_name)
print(result)
```

OUTPUT:

```
Temperature: 25.47°C
Description: Few clouds
```

CONCLUSION:

- This live weather app provides a user-friendly and comprehensive way to access real-time weather information for any location, empowering you to stay informed and prepared for the ever-changing elements
- The live weather app successfully displays real-time weather information for any location, providing users with accurate and up-to-date data to make informed decisions.
- "Never be caught off guard by the weather again. This app provides essential weather updates, ensuring you're always prepared for what's to come."
- "For those who travel frequently, the multi-location support makes keeping track of weather across different cities a breeze."
- The project successfully integrated with a reliable weather API and utilized effective data parsing techniques to present weather information in a clear and easy-to-understand manner.