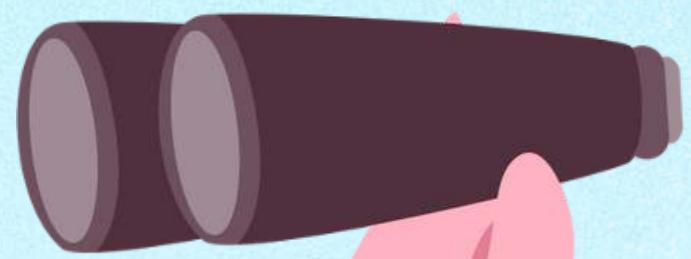


WEATHER FORECAST

Mathen
Firdaus

U2102842
U2001991



PURPOSE

To provide users with current weather conditions such as temperature, humidity, wind speed, and overall weather description

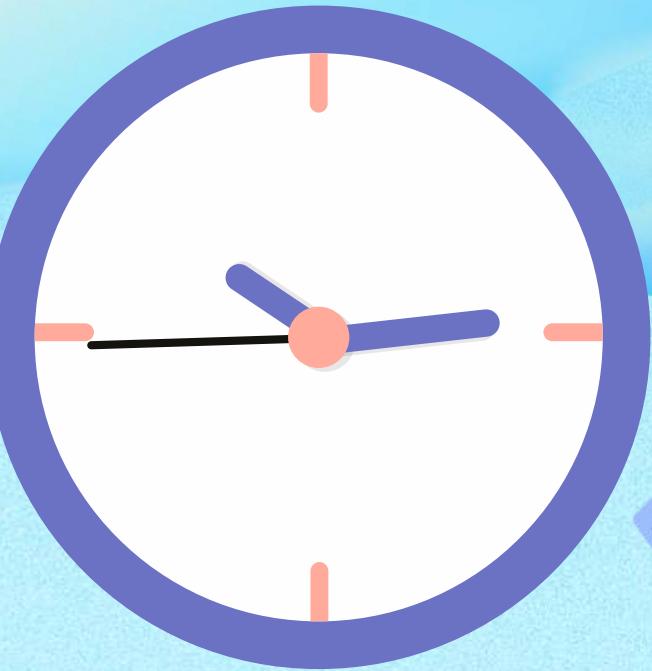
provide a simple interface for users to check weather conditions by entering their location.

API



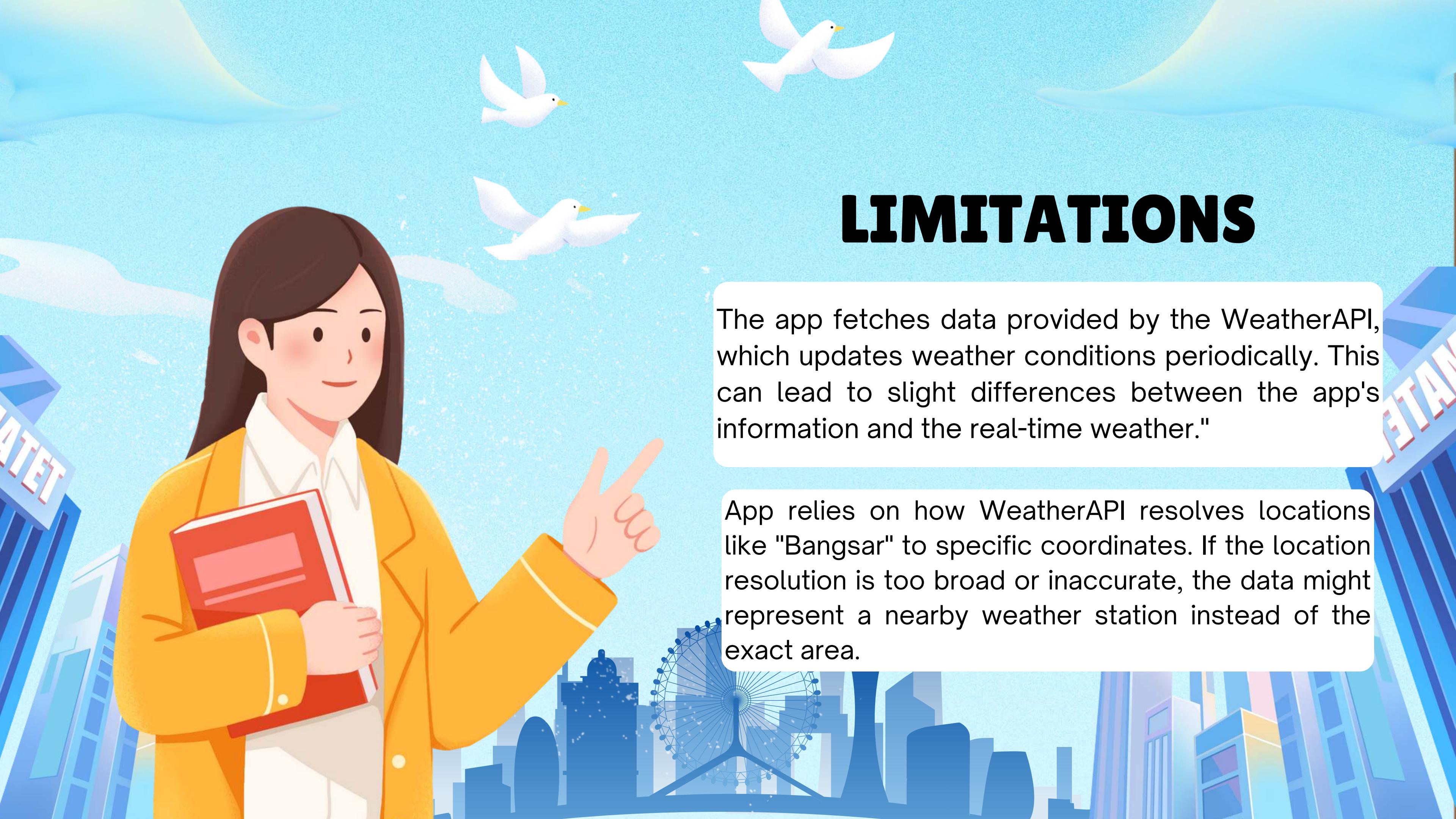
WeatherAPI offers real-time weather data as well as forecasts.

Allows developers to integrate weather-related information into their applications, websites, or services.



TETAM

LIMITATIONS



The app fetches data provided by the WeatherAPI, which updates weather conditions periodically. This can lead to slight differences between the app's information and the real-time weather."

App relies on how WeatherAPI resolves locations like "Bangsar" to specific coordinates. If the location resolution is too broad or inaccurate, the data might represent a nearby weather station instead of the exact area.

API USAGE

```
API_KEY = "0f5602131ed3443d9dc152340242012" # Replace with your WeatherAPI key
BASE_URL = "http://api.weatherapi.com/v1"

# Function to fetch weather data
def get_weather_data(location):
    url = f"{BASE_URL}/forecast.json"
    params = {
        "key": API_KEY,
        "q": location,
        "days": 3, # Fetch data for 3 days
        "aqi": "no",
        "alerts": "no"
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"API call failed: {response.status_code}, {response.text}")

# Function to parse weather data
def parse_weather_data(data):
    current = data["current"]
    forecast = data["forecast"]["forecastday"]

    current_weather = {
        "temperature": current["temp_c"],
        "condition": current["condition"]["text"],
        "humidity": current["humidity"],
        "wind_speed": current["wind_kph"]
    }

    forecast_data = []
    for day in forecast:
        forecast_data.append({
            "date": day["date"],
            "max_temp": day["day"]["maxtemp_c"],
            "min_temp": day["day"]["mintemp_c"],
            "condition": day["day"]["condition"]["text"]
        })
    return {"current": current_weather, "forecast": forecast_data}
```

Call the API

- Use the BASE_URL (<http://api.weatherapi.com/v1>) and specify the endpoint (/forecast.json).
- Include the necessary parameters in the query:
 - "key": Your API key.
 - "q": The location for which weather data is required.
 - "days": The number of days of forecast data you want to retrieve.

Additional parameters like "aqi" (air quality) and "alerts" can be included to refine the request.

Make the API Call:

- Use the requests library to send a GET request to the WeatherAPI endpoint, passing the parameters in the URL.

For example: `requests.get(url, params=params)`.

Handle Responses:

- Check if the response status code is 200 (OK) to ensure that the API call was successful.

Parse the returned JSON data using `response.json()` to extract the weather information.

Process Data:

- Extract and format the relevant data from the JSON response:
- Current weather: Temperature, humidity, wind speed, and conditions.
- Forecast data: Max and min temperatures, forecast conditions for the specified days.
- Handle any potential errors or missing fields gracefully (e.g., in case of missing data or API failure).

THANK YOU

