

Weather App

Abstract

This project mainly focuses on real-time weather information, implementing a user-friendly interface and incorporating data analysis for enhanced user experience.

1 Introduction

1.1 Background

Weather App aims to leverage Python's capabilities to create a user-friendly weather app, catering to the increasing demand for accurate and accessible weather information.

1.2 Problem Statement

Existing weather applications lack certain functionalities or present data in a complex manner. Weather App addresses these issues, providing a streamlined solution for users.

1.3 Objectives:

- Develop a Python-based weather application with a clean interface.
- Implement real-time weather data retrieval.

2 Methodology

2.1 Tools and Technologies Used:

Weather App utilizes Python programming language, incorporating libraries such as Requests for API calls and Tkinter for the graphical user interface(GUI).

2.2 Project Design:

The project design encompasses a clear data flow, utilizing algorithms to process weather data efficiently. The architectural design ensures scalability and modularity.

2.3 Implementation Details:

Key Python functions and classes are employed to handle data retrieval, processing, and display. The code focuses on readability and maintainability.

3 Results and Discussion

3.1 Project Outcomes:

Weather App successfully delivers real-time weather information through an intuitive interface, empowering users with accurate and accessible data.

3.2 Challenges Faced:

Challenges included API integration and optimizing data retrieval processes. Overcoming these challenges strengthened the project's robustness.

3.3 Learnings and Insights:

Working on Weather App provided insights into effective API usage, data analysis, and graphical user interface design using Python.

4 Conclusion

Weather App presents a viable solution for users seeking a Python-based weather application. Its impact lies in simplifying weather information retrieval and analysis. Future work includes refining UI elements and expanding feature sets.

References:

Open weather API, Tkinter Module, Request Module, Json Module, Date Time Module, PIL Module

Appendices

For some more details we have attached an Image of the Code as well as of the Output.

Code Snippet

```
#Condition for data fetching
if weather_info['cod'] == 200:
    kelvin = 273

    # Storing the fetched values of weather of a city

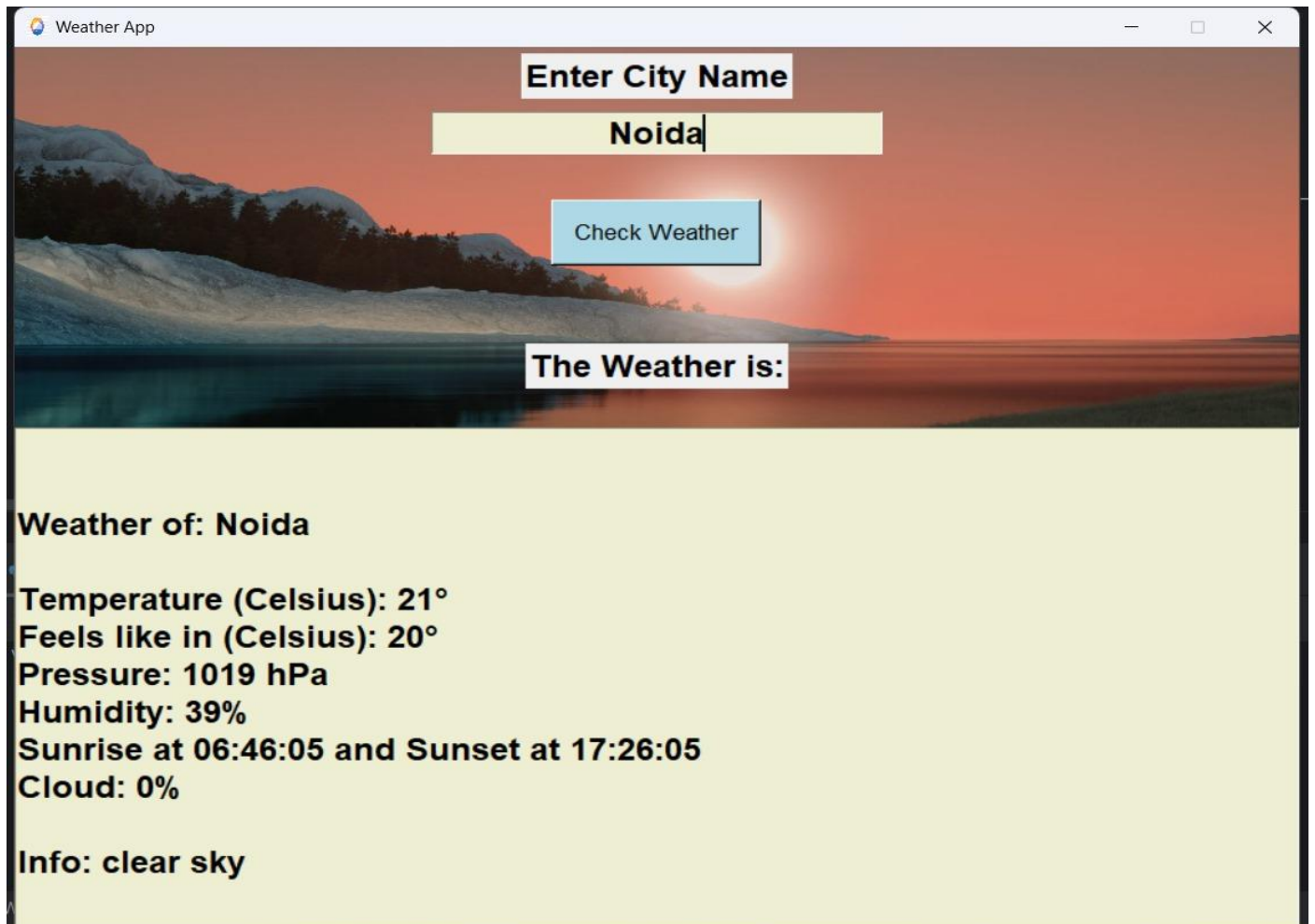
    temp = int(weather_info['main']['temp'] - kelvin)
    feels_like_temp = int(weather_info['main']['feels_like'] - kelvin)
    pressure = weather_info['main']['pressure']
    humidity = weather_info['main']['humidity']
    wind_speed = weather_info['wind']['speed'] * 3.6
    sunrise = weather_info['sys']['sunrise']
    sunset = weather_info['sys']['sunset']
    timezone = weather_info['timezone']
    cloudy = weather_info['clouds']['all']
    description = weather_info['weather'][0]['description']

    sunrise_time = time_format_for_location(sunrise + timezone)
    sunset_time = time_format_for_location(sunset + timezone)

    weather = (f"\n\nWeather of: {city_name}\n\nTemperature (Celsius): {temp}°\nFeels like in (Celsius): {feels_like_temp}°\nPressure: {pressure} hPa\n"
              f"Humidity: {humidity}%\nSunrise at {sunrise_time} and Sunset at {sunset_time}\nCloud: {cloudy}%\n\nInfo: {description}")
else:
    weather = f"\n\nWeather for '{city_name}' not found!\n\nKindly Enter valid City Name !!"

tfield.insert(INSERT, weather)
```

Output Snippet



The screenshot shows a web browser window with the title 'Weather App'. The background of the page is a scenic image of a lake at sunset with mountains in the distance. The interface includes a text input field with the label 'Enter City Name' containing the text 'Noida', a blue button labeled 'Check Weather', and a section titled 'The Weather is:' which contains the following weather data:

Weather of: Noida
Temperature (Celsius): 21°
Feels like in (Celsius): 20°
Pressure: 1019 hPa
Humidity: 39%
Sunrise at 06:46:05 and Sunset at 17:26:05
Cloud: 0%
Info: clear sky

THANK

YOU!!