**Project Name:** Weather App **Programming Language:** Python

**Description:**

Create the main window

root = tk.Tk() root.config(bg = "sky blue") root.geometry("500x500") root.title("Weather App")

label = tk.Label(root, text = "Weather App", font = ('Arial', 25)) label.pack(padx=25, pady=25)

textbox = tk.Text(root, font = ('Arial', 14)) textbox.place(x=80, y=90, height=50, width=350)

label\_T = tk.Label(root, text = "Temperature", font = ('Arial', 25)) label\_T.place(x=15, y=220, height=50, width=210) label\_T1 = tk.Label(root, text = "", font = ('Arial', 25)) label\_T1.place(x=240, y=220, height=50, width=210)

label\_H = tk.Label(root, text = "Humidity", font = ('Arial', 25)) label\_H.place(x=15, y=280, height=50, width=210) label\_H1 = tk.Label(root, text = "", font = ('Arial', 25)) label\_H1.place(x=240, y=280, height=50, width=210)

label\_W = tk.Label(root, text = "Pressure", font = ('Arial', 20)) label\_W.place(x=15, y=340, height=50, width=210) label\_W1 = tk.Label(root, text = "", font = ('Arial', 20)) label\_W1.place(x=240, y=340, height=50, width=210)

label\_S = tk.Label(root, text = "Wind Speed", font = ('Arial', 20)) label\_S.place(x=15, y=400, height=50, width=210) label\_S1 = tk.Label(root, text = "", font = ('Arial', 20)) label\_S1.place(x=240, y=400, height=50, width=210)

button = tk.Button(root, text="Check Weather", font=('Arial', 15), command=fetch\_and\_display\_weather) button.pack(padx=50, pady=60)

Start the Tkinter event loop root.mainloop()

make a Documentation for this code

Edit

Weather App Documentation

Overview

This is a Python script that creates a GUI-based weather app using Tkinter. The app allows users to enter a location and retrieve the current weather data, including temperature, humidity, pressure, and wind speed. The data is fetched from the OpenWeatherMap API and displayed in the app.

Classes

Weather

\_\_init\_\_: Initializes a Weather object with temperature, humidity, pressure, and wind speed attributes. display\_weather: Abstract method to display the weather data. save\_to\_csv: Saves the weather data to a CSV file.

WeatherAPI

\_\_init\_\_: Initializes a WeatherAPI object with an API key.

fetch\_weather: Abstract method to fetch weather data from an API.

CurrentWeather

\_\_init\_\_: Initializes a CurrentWeather object with temperature, humidity, pressure, and wind speed attributes. display\_weather: Displays the current weather data.

OpenWeatherAPI

\_\_init\_\_: Initializes an OpenWeatherAPI object with an API key. fetch\_weather: Fetches weather data from the OpenWeatherMap API. UserInput get\_location: Static method to get a location input from the user. WeatherDisplay show: Static method to display the weather data.

Functions

fetch\_and\_display\_weather

Fetches the weather data for the entered location and displays it in the app.

GUI Components

root: The main window of the app. label: A label displaying the app title. textbox: A text box for entering the location.

label\_T, label\_H, label\_W, label\_S: Labels displaying the weather data headers.

label\_T1, label\_H1, label\_W1, label\_S1: Labels displaying the weather data values. button: A button to trigger the weather data fetch and display.

API Key

api\_key: The API key for the OpenWeatherMap API.

CSV File

weather\_data.csv: The file where the weather data is saved.

How to Use:

Run the script to launch the app.

Enter a location in the text box.

Click the "Check Weather" button to fetch and display the weather data.

The weather data will be displayed in the app and saved to the weather\_data.csv file.

Edit