WEATHER REPORT

A MINI PROJECT REPORT

18CSC207J - ADVANCED PROGRAMMING PRACTICE

Submitted by

KARUMANCHI ROSHAN [RA211100300779]
CH CHETHAN RAJ [RA2111003010772]
VUNADI RISHI VARDHAN REDDY [RA2111003010786]
MEESALA ARAVIND [RA2111003010795]

Under the guidance of

Ms.Nithiya

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"WEATHER REPORT"** is the bona fide work of **KARUMANCHI ROSHAN(RA2111003010779)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

GUIDE NAME
Ms.Nithiya
Assistant Professor
Department of Computing Technologies

ABSTRACT

The project is a simple weather application that allows the user to get weather information for a selected city. The application is developed using the Python programming language and the Tkinter library is used for creating the graphical user interface (GUI).

The application uses the OpenWeatherMap API to fetch the weather data for the selected city. The API provides a JSON response, which is parsed to extract the required information such as weather status, temperature, humidity, and wind speed.

The GUI of the application consists of a dropdown list containing the names of various cities. The user can select a city from the list and click the "Get Weather" button to fetch the weather information for that city. The weather information is displayed in the form of labels, which are dynamically updated based on the selected city.

The application is designed to be simple and easy to use, with a clean and intuitive interface. It provides a quick and convenient way for users to check the weather for a particular city. The project can be further extended by adding more features such as a forecast for multiple days, weather alerts, etc.

MODULES

This code is a Python program that creates a simple weather application using the Tkinter module for the graphical user interface and the Requests module for making HTTP requests to the OpenWeatherMap API.

The program first defines a function called weather() that gets the city name from the drop-down listbox, constructs a URL with the city name and the OpenWeatherMap API key, sends a GET request to the API using the Requests module, and extracts relevant weather information from the response JSON data. The weather information includes the weather status, temperature, humidity, and wind speed. The function then updates the weather status, temperature, humidity, and wind speed labels with the extracted information.

The program then creates a Tkinter window with a title, size, and background color. It also creates a drop-down listbox with a list of city names, a "Get Weather" button, and four labels for displaying weather information. The program sets the default value of the drop-down listbox to "Select City" and sets the color of the button and labels.

When the user selects a city from the drop-down listbox and clicks the "Get Weather" button, the program calls the weather() function to get and display the weather information for the selected city.

CODE

```
import requests
from tkinter import *
def weather():
    city = city_listbox.get()

url="https://api.openweathermap.org/data/2.5/weather?q={}&appid=62f668dc94069b9a16445e8a77559b6e".format(city)
    res = requests.get(url)
    output = res.json()

    weather_status = output['weather'][0]['description']
    temprature = output['main']['temp']
    humidity = output['main']['humidity']
    wind_speed = output['wind']['speed']

    weather_status_label.configure(text="Weather Status: " + weather_status.title(), fg="#FFFFFF", bg="#5D5D5D")
    temprature_label.configure(text="Temprature: " + str(round(temprature - 273.15, 2)) + " °C", fg="#FFFFFF",            bg="#5D5D5D")
    humidity_label.configure(text="Humidity: " + str(humidity) + " %", fg="#FFFFFF", bg="#5D5D5D")
    wind_speed_label.configure(text="Wind Speed: " + str(wind_speed) + " km/hr", fg="#FFFFFF", bg="#5D5D5D")

window = Tk()
window.title("Weather App")
window.geometry("400x350")
```

```
window.minsize(300, 250)

# Adding Background Color
bg_color = "#6CAEDF"
window.config(bg=bg_color)

city_name_list = ["Delhi", "Mumbai","", "Kolkata","Eluru","Kannur","Chennai", "Bangalore",
"Hyderabad", "Pune", "Ahmedabad", "Jaipur", "Lucknow", "Kanpur", "Nagpur", "Patna",
"Bhopal", "Visakhapatnam", "Vadodara", "Coimbatore", "Ludhiana", "Nashik", "Agra",
"Faridabad", "Meerut", "Rajkot", "Varanasi", "Srinagar", "Allahabad", "Jabalpur", "Amritsar",
"Gwalior", "Vijayawada", "Madurai", "Jodhpur", "Raipur", "Kota", "Guwahati", "Chandigarh",
"Solapur", "Hubli-Dharwad", "Bareilly", "Moradabad", "Mysore", "Gurgaon", "Aligarh",
"Jalandhar", "Tiruchirappalli", "Bhubaneswar", "Salem", "Mira-Bhayandar",
"Thiruvananthapuram", "Bhiwandi", "Saharanpur", "Gorakhpur", "Guntur", "Bikaner",
"Amravati", "Noida", "Jamshedpur", "Bhilai", "Warangal", "Cuttack", "Firozabad", "Kochi",
"Bhavnagar", "Dehradun", "Durgapur", "Asansol", "Nanded", "Kolhapur", "Ajmer",
"Gulbarga", "Jamnagar", "Ujjain", "Loni", "Siliguri", "Jhansi", "Ulhasnagar", "Nellore",
"Jammu", "Sangli-Miraj & Kupwad", "Belgaum", "Mangalore", "Ambattur", "Tirunelveli",
"Malegaon", "Gaya", "Jalgaon", "Udaipur", "Maheshtala", "Davanagere", "Kozhikode",
"Akola", "Kurnool", "Rajpur Sonarpur", "Rajahmundry", "Bokaro", "South Dumdum",
"Bellary", "Patiala", "Gopalpur", "Agartala", "Bardhaman", "Barasat", "Bijapur",
"Shahjahanpur", "Kakinada", "Sikar", "Kollam", "Tumkur", "Kharagpur"]
city_name_list.sort()

city_listbox = StringVar(window)
city_listbox.set("Select City")
city_listbox_menu = OptionMenu(window, city_listbox, *city_name_list)
city_listbox_menu.config(width=15, relief=FLAT)
city_listbox_menu.grid(row=2, column=2, padx=20, pady=10, sticky="ew")

# Adding Button Color
```

```
button_color = "#3B3B3B"
b1 = Button(window, text="Get Weather", width=12, command=weather, bg=button_color,
fg="#FFFFFF")
b1.grid(row=5, column=2, padx=20, pady=10, sticky="n")

# Adding Labels and Setting their Colors
weather_status_label = Label(window, font=("Helvetica", 12, "bold"))
weather_status_label.grid(row=10, column=2, padx=20, pady=10, sticky="n")

temprature_label = Label(window, font=("Helvetica", 12, "bold"))
temprature_label.grid(row=12, column=2, padx=20, pady=10, sticky="n")

humidity_label = Label(window, font=("Helvetica", 12, "bold"))
humidity_label.grid(row=14, column=2, padx=20, pady=10, sticky="n")

wind_speed_label = Label(window, font=("Helvetica", 12, "bold"))
wind_speed_label.grid(row=16, column=2, padx=20, pady=10, sticky="n")

# Making Widgets Flexible
window.grid_columnconfigure(2, weight=1)
window.grid_rowconfigure(1, weight=1)
window.grid_rowconfigure(3, weight=1)

window.mainloop()
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [1]:

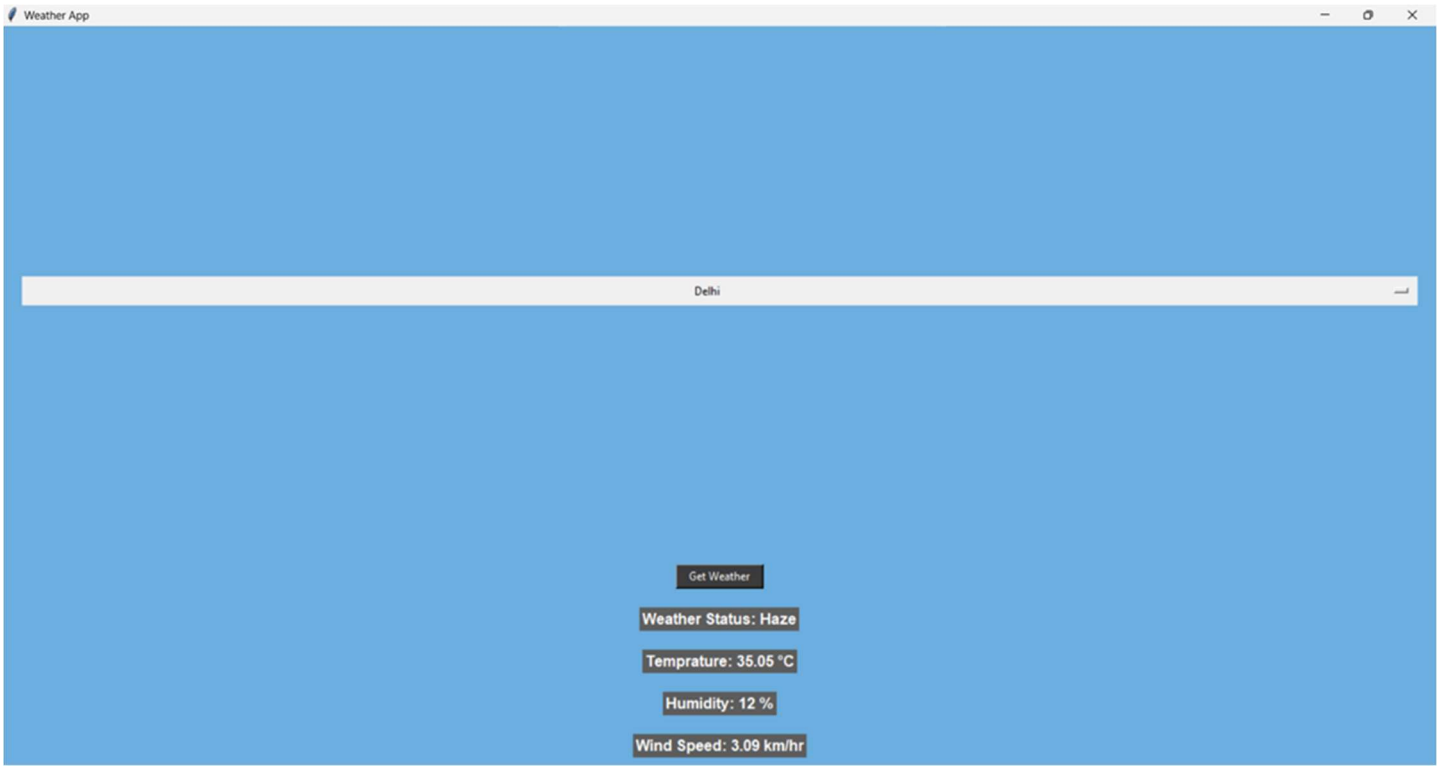
```
import requests
from tkinter import *

def weather():
    city = city_listbox.get()
    url = "https://api.openweathermap.org/data/2.5/weather?q={}&appid=62f668dc94069b9a16445e8a77559b6e".format(city)
    res = requests.get(url)
    output = res.json()

    weather_status = output['weather'][0]['description']
    temprature = output['main']['temp']
    humidity = output['main']['humidity']
    wind_speed = output['wind']['speed']

    weather_status_label.configure(text="Weather Status: " + weather_status.title(), fg="#FFFFFF", bg="#5D5D5D")
    temprature_label.configure(text="Temprature: " + str(round(temprature - 273.15, 2)) + " °C", fg="#FFFFFF", bg="#5D5D5D")
    humidity_label.configure(text="Humidity: " + str(humidity) + " %", fg="#FFFFFF", bg="#5D5D5D")
    wind_speed_label.configure(text="Wind Speed: " + str(wind_speed) + " km/hr", fg="#FFFFFF", bg="#5D5D5D")

window = Tk()
window.title("Weather App")
window.geometry("400x350")
window.minsize(300, 250)

# Adding Background Color
bg_color = "#6CAEDF"
window.config(bg=bg_color)

city_name_list = ["Delhi", "Mumbai","", "Kolkata","Eluru","Kannur","Chennai", "Bangalore", "Hyderabad", "Pune", "Ahmedabad", "Jai
city_name_list.sort()

city_listbox = StringVar(window)
city_listbox.set("Select City")
city_listbox_menu = OptionMenu(window, city_listbox, *city_name_list)
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
city_name_list = ["Delhi", "Mumbai","", "Kolkata","Eluru","Kannur","Chennai", "Bangalore", "Hyderabad", "Pune", "Ahmedabad", "Jai
city_name_list.sort()

city_listbox = StringVar(window)
city_listbox.set("Select City")
city_listbox_menu = OptionMenu(window, city_listbox, *city_name_list)
city_listbox_menu.config(width=15, relief=FLAT)
city_listbox_menu.grid(row=2, column=2, padx=20, pady=10, sticky="ew")

# Adding Button Color
button_color = "#3B3B3B"
b1 = Button(window, text="Get Weather", width=12, command=weather, bg=button_color, fg="#FFFFFF")
b1.grid(row=5, column=2, padx=20, pady=10, sticky="n")

# Adding Labels and Setting their Colors
weather_status_label = Label(window, font=("Helvetica", 12, "bold"))
weather_status_label.grid(row=10, column=2, padx=20, pady=10, sticky="n")

temprature_label = Label(window, font=("Helvetica", 12, "bold"))
temprature_label.grid(row=12, column=2, padx=20, pady=10, sticky="n")

humidity_label = Label(window, font=("Helvetica", 12, "bold"))
humidity_label.grid(row=14, column=2, padx=20, pady=10, sticky="n")

wind_speed_label = Label(window, font=("Helvetica", 12, "bold"))
wind_speed_label.grid(row=16, column=2, padx=20, pady=10, sticky="n")

# Making Widgets Flexible
window.grid_columnconfigure(2, weight=1)
window.grid_rowconfigure(1, weight=1)
window.grid_rowconfigure(3, weight=1)

window.mainloop()
```


Weather App
Select City
Get Weather


Weather App
Delhi
Get Weather
Weather Status: Haze
Temprature: 35.05 °C
Humidity: 12 %
Wind Speed: 3.09 km/hr

CONCLUSION

In conclusion, This is a Python program that creates a GUI application for checking the weather of a city using the OpenWeatherMap API. It imports the requests module to send an HTTP request to the OpenWeatherMap API and retrieve the weather data in JSON format. It also imports the tkinter module to create the GUI interface.

The `weather()` function is defined to get the weather data of the selected city from the OpenWeatherMap API and display it in the GUI. It gets the city name from the `city_listbox` widget, constructs the URL for the API request, sends the request, and extracts the relevant weather data from the JSON response. It then updates the weather status, temperature, humidity, and wind speed labels with the retrieved data.

The `window` object is created with the `Tk()` constructor and various attributes such as the title, geometry, and background color are set. The list of city names is sorted alphabetically and stored in the `city_name_list` variable. The `city_listbox` widget is created with the `StringVar()` constructor and set to the default value "Select City". An `OptionMenu()` widget is created with the sorted list of city names as options, and the `city_listbox` variable is set as its initial value. The "Get Weather" button is created with the `Button()` constructor and set to call the `weather()` function when clicked. The weather status, temperature, humidity, and wind speed labels are created with the `Label()` constructor and positioned on the GUI.

Overall, this program provides a simple and user-friendly way for users to check the weather of their desired city in real-time.