पी.एम. श्री. केन्द्रीय विद्यालय सुंदरगढ़

PM SHRI KENDRIYA VIDYALAYA SUNDARGARH

PROJECT ON:-
WEATHER APP WITH PYTHON

SUBJECT:-COMPUTER SCIENCE

SUBMITTED BY:-

NAME: R JAYAM
STD: XII(SCIENCE)
ROLL NO:

GUIDED BY:- MR. SATISH KUMAR EKKA
(PGT COMPUTER SCIENCE)

CERTIFICATE

This is to certify that R.JAYAM of Class XII science has prepared the investigatory **Computer Science** project entitled "**WEATHER APP WITH PYTHON**". The report is found worthy of acceptance as annual project report for the subject Computer Science of Class XII. He has prepared the report under my guidance.

Signature of Principal

Signature of Internal examiner

Signature of External Examiner

DECLARATION

I hereby declare that investigatory project entitled "**Weather App With python**", is a bona fide work of me and carried out successfully in a partial fulfilment of computer science project carried out during the session 2023-24.

Name: R Jayam
Class: XII Science
Roll no:

____
Signature of Student

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks & gratitude to my project guide MR. SATISH KUMAR EKKA for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & constant motivation have been responsible for the successful completion of this project.

My sincere thanks go to MR. NAVNEET AMKHARE, Our principal sir , for his co-ordination in extending every possible support for the completion of this project.

I must thanks to my classmates for their timely help & support for compilation of this project.

Finally, I would like to thank the CBSE board for giving me this great opportunity to do this project.

Signature of Student

CONTENTS

INTRODUCTION

Our class 12 computer science project introduces a Weather App developed using Python, bridging the worlds of computer science and meteorology. This Weather App offers real-time weather updates and forecasts, making it an essential tool in our digital age. By leveraging Python's versatility, we have created an interactive application that grants users easy access to comprehensive weather data. In this project, we will explore data management, web scraping, API integration, and GUI development. We aim to enhance our coding skills while highlighting the practical application of computer science in our daily lives. The app's features include location-based weather data, forecasting, detailed information, and an intuitive GUI. Data accuracy is crucial, and we prioritize reliable sources. Future improvements may include geolocation services, severe weather alerts, historical data access, and user customization. Our Weather App represents a significant achievement, exemplifying the synergy of technology and meteorology, and promises an exciting and educational journey into the world of Python programming.

SOFTWARE REQUIREMENTS

- **WORK IN HARDWARE AND SOFTWRE:-**
  - IDLE python and Pycharm
  - Windows11
  - I5-1240p 12thGen
  - 8GB Ram
  - Intel(R) Iris(R)Xe Graphics
  - MySQL

- **Minimum System requirement:-**
  - Windows10
  - I3 10th Gen
  - 4GB Ram

PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages.

Here some Features of Python are:-

- Dynamically typed
- Free and open-source software
- High-level programming language
- Interpreted
- Object-oriented programming
- Standard library
- Easy to code and Learn

MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My and "SQL", the acronym for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, Php ,BB, and WordPress.[10] MySQL is also used by many popular websites, including Facebook, Flickr, Media Wiki, Twitter, and YouTube.

WEATHER API

A weather API, also known as a weather data API, is a web service that provides access to various meteorological data and weather-related information. Developers and applications use these APIs to retrieve current weather conditions, weather forecasts, historical weather data, and other meteorological information. Weather APIs are valuable for a wide range of applications, including weather apps, websites, IoT devices, and more.

I use OpenWeatherMap API . It’s offers a comprehensive weather API that provides access to current weather data, forecasts, historical weather data, and other weather-related information. It covers a wide range of geographical locations and provides data on temperature, humidity, wind speed, precipitation, and more.

When using a weather API, you typically need to sign up for an API key, which is a unique identifier that allows you to access the API's services. You make HTTP requests to the API's endpoints, passing your API key and any necessary parameters, and receive weather data in response, usually in JSON format. These APIs may have free tiers with limited requests and paid plans for higher usage and additional features.

WHY DO WE NEED IT?

A weather app developed with Python can offer several advantages and fulfil various needs, contributing to its relevance and usefulness. Here are some reasons why a weather app with Python is beneficial: Real-time Updates: Python allows for seamless integration with APIs that provide real-time weather data. This ensures that users receive the most current and accurate information about the weather conditions in their location or any specified location. Versatility: Python is a versatile programming language that can be used for both web and desktop application development. This versatility allows developers to create weather apps that are accessible across different platforms, including desktop computers, mobile devices, and the web. Rich Ecosystem of Libraries: Python has a rich ecosystem of libraries and frameworks that facilitate the development of feature-rich applications. Libraries such as Requests, Beautiful Soup, and Matplotlib can be employed to fetch and visualize weather data effectively. User Customization: Python's flexibility enables developers to create weather apps with customizable features. Users can personalize their experience by selecting preferred units of measurement, setting location preferences, and receiving notifications for specific weather events. Integration of Geolocation Services: Python allows for seamless integration of geolocation services, enabling the app to determine the user's location automatically. This feature enhances user convenience, providing localized weather information without manual input. Data Analysis and Visualization: Python's data analysis and visualization capabilities, through libraries like Pandas and Matplotlib, allow developers to present weather trends, historical data, and forecasts in a visually appealing and comprehensible manner. Cross-platform Compatibility: Python's cross-platform compatibility ensures that the weather app can be deployed on various operating systems without significant modifications. This enhances accessibility and usability for a broader audience. Community Support: Python has a large and active community of developers, which means that there are plenty of resources, tutorials, and third-party libraries available to aid in the development of the weather app. This community support can expedite the development process and troubleshoot issues more efficiently.

OBJECTIVE

Develop a comprehensive and user-friendly weather application using Python, aimed at providing real-time and accurate weather information to users. The primary goal is to create a versatile application that offers a seamless user experience while leveraging various APIs and Python libraries to gather, process, and display weather data. This weather app will not only showcase the current weather conditions but also include features such as forecasts, historical data, and user customization options. The project aims to enhance user engagement by implementing an intuitive graphical interface, ensuring cross-platform compatibility, and integrating advanced features, such as geolocation services and notification alerts. Through this project, the objective is to empower users with a reliable tool that facilitates informed decision-making based on up-to-date and relevant weather information.

MODULE USED

In This project module used:-

- Tkinter
- Request
- Pytz
- Geopy
- Datetime
- Timezonefinder
- Auto-py-to-exe

NOTE:-For Installing Modules Type command on CMD <pip install (Module name)>

Tkinter:-

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app*:*

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Request:-

Requests library is one of the integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scraping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

Pytz:-

Pytz brings the Olson tz database into Python and thus supports almost all time zones. This module serves the date-time conversion functionalities and helps user serving international client's base. It enables time-zone calculations in our Python applications and also allows us to create timezone aware datetime instances.

Geopy:-

Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

Datetime:-

It is use for date and time in python.

Timezonefinder:-

The Timezonefinder module is able to find the timezone of any point on earth (coordinates) offline.

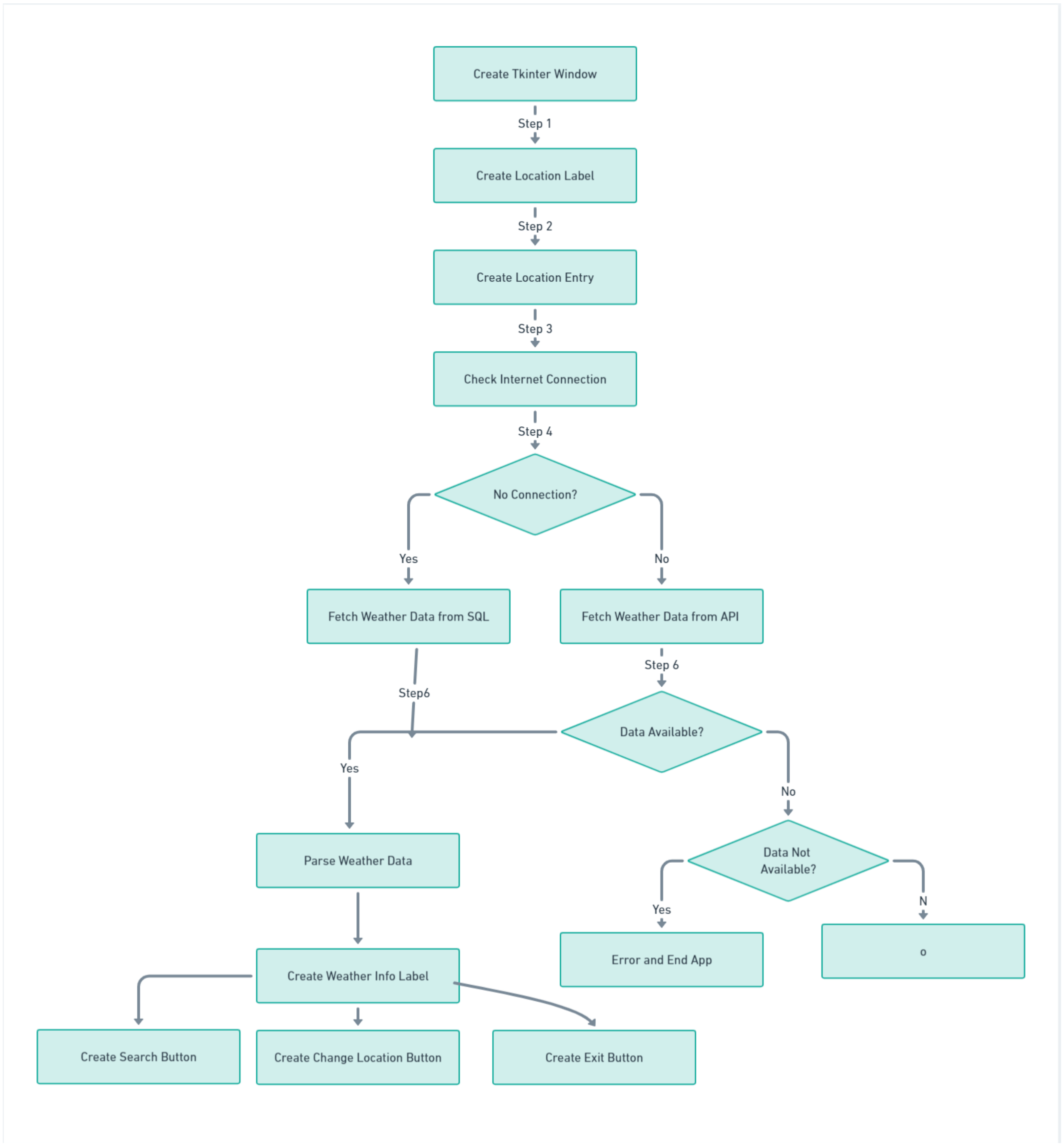
Auto-py-to-exe:-

Auto-py-to-exe is a graphical user interface (GUI) tool for converting Python scripts into standalone executable files for Windows. It simplifies the process of packaging your Python code and its dependencies into a single executable file that can be run on a Windows system without the need for a Python interpreter. This tool is especially useful for sharing your Python applications with users who may not have Python installed on their machines.

MYSQL Connector:-

Python needs a MySQL driver to access the MySQL database.
In this tutorial we will use the driver "MySQL Connector".
We recommend that you use PIP to install "MySQL Connector".
PIP is most likely already installed in your Python environment.

FLOWCHART


Create Tkinter Window
Step 1
Create Location Label
Step 2
Create Location Entry
Step 3
Check Internet Connection
Step 4
No Connection?
Yes
No
Fetch Weather Data from SQL
Fetch Weather Data from API
Step 6
Step6
Data Available?
Yes
No
Parse Weather Data
Data Not Available?
Yes
N
Error and End App
o
Create Weather Info Label
Create Search Button
Create Change Location Button
Create Exit Button

- **Here is a step-by-step breakdown of the Weather App flow:**
    1. Create Tkinter Window: The GUI (Graphical User Interface) window is created with the title "Weather App".
    2. Create Location Label: A label is added to the window with the text "Enter Location:".
    3. Create Location Entry: An entry field is added to the window where the user can input their desired location.
    4. Check Internet Connection: The app checks if there is an active internet connection.
    5. If No Connection: If there is no internet connection ,Fetch data from SQL or an error label is created with the text "No Internet Connection", and the app ends.
    6. Fetch Weather Data from API: If there is an internet connection, the app fetches weather data from an API.
    7. If Data Not Available: If the weather data is not available or there is an error in fetching it, an error label is created with the text "Data Unavailable", and the app ends.
    8. Parse Weather Data: The app parses the fetched weather data to extract the relevant information.
    9. Create Weather Info Label: A label is created to display the weather information obtained from the parsed data.
    10. Create Search Button: A button labelled "search" is added to the window, which search the weather data locations when clicked.
    11. Create Change Location Button: A button labelled "Change Location" is added to the window, which triggers the on change location() function when clicked.
    12. Create Exit Button: A button labelled "Exit" is added to the window, which ends the app when clicked.

CODING

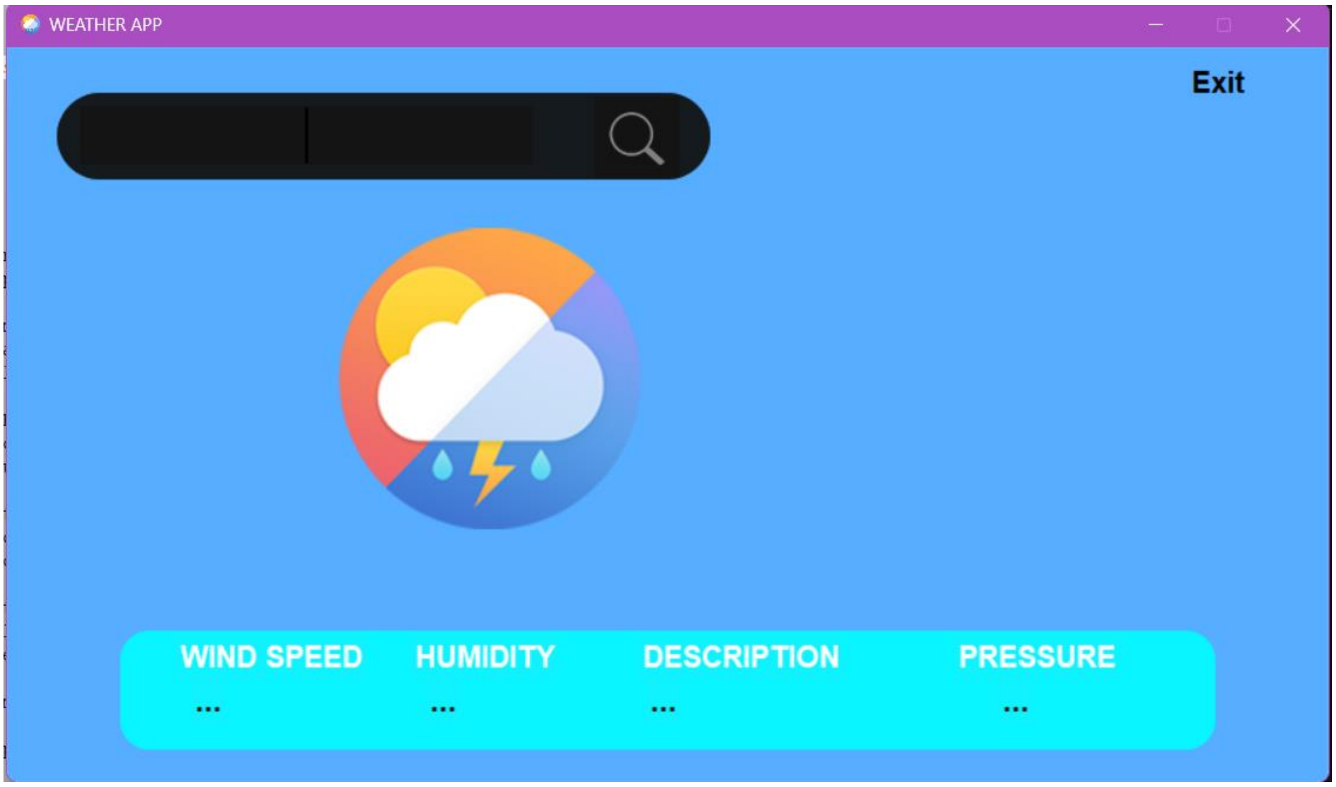
```
from tkinter import *
import tkinter as tk
from geopy.geocoders import Nominatim
from tkinter import ttk, messagebox
from timezonefinder import TimezoneFinder
from datetime import datetime
import requests
import pytz
import mysql.connector as mycon

root=Tk()
root.title("WEATHER APP")
root.geometry("900x500")
root.configure(bg="#57adff")
root.resizable(False,False)

def exit_application():
    if messagebox.askokcancel("Exit", "Do you really want to exit?"):
        root.destroy()

def getweather():
   try:
       city = textf.get()

       getloc = Nominatim(user_agent="geopiExercise")
       location = getloc.geocode(city)
       obj = TimezoneFinder()
       result = obj.timezone_at(lng=location.longitude, lat=location.latitude)

       timezone.config(text=result)
       home = pytz.timezone(result)
       localt = datetime.now(home)
       current_t = localt.strftime("%I:%M:%p")
       clock.config(text=current_t)
       current_d= localt.strftime("%d-%m-%Y")
       date1.config(text=current_d)
       long1.config(text=f"{round(location.latitude,
4)}°N,{round(location.longitude, 4)}°E")
       name.config(text="CURRENT WEATHER")
```

```
        api = "https://api.openweathermap.org/data/2.5/weather?q=" + city +
"&appid=1be6666b3f28e54c35779e4ca6a28f42"

        json_d = requests.get(api).json()
        con = json_d['weather'][0]['main']
        des = json_d['weather'][0]['description']
        temp = int(json_d['main']['temp'] - 273.15)
        pre = json_d['main']['pressure']
        hum = json_d['main']['humidity']
        wind = json_d['wind']['speed']

        t.config(text=(temp, "°"))
        c.config(text=(con, "|", "FEELS", "LIKE", temp, "°"))
        w.config(text=(wind,"m/s"))
        h.config(text=(hum,"%"))
        d.config(text=des)
        p.config(text=(pre,"hPa"))


    except Exception as e:
        city = textf.get()
        mcon = mycon.connect(host='127.0.0.1', user='root', password="root")
        cur = mcon.cursor()
        cur.execute("use weather")
        mcon.commit()
        con = f"Select Predictions from data where D_S_name='{city}'"
        cur.execute(con)
        conr = cur.fetchone()

        des = f"Select description from data where D_S_name='{city}'"
        cur.execute(des)
        desr = cur.fetchone()

        temp = f"Select temp from data where D_S_name='{city}'"
        cur.execute(temp)
        tempr = cur.fetchone()

        pre = f"Select pressure from data where D_S_name='{city}'"
        cur.execute(pre)
        prer = cur.fetchone()
        hum = f"Select humidity from data where D_S_name='{city}'"
        cur.execute(hum)
        humr = cur.fetchone()

        wind = f"Select wind_speed from data where D_S_name='{city}'"
        cur.execute(wind)
        windr = cur.fetchone()

        if desr==None:
            messagebox.showerror("Weather App", ""Sorry! Data is not available
\n It will be available Soon!!")
        else:
            t.config(text=(tempr, "°"))
            c.config(text=(conr, "|", "FEELS", "LIKE", tempr, "°"))
            w.config(text=(windr, "m/s"))
            h.config(text=(humr, "%"))
            d.config(text=desr)
            p.config(text=(prer, "hPa"))

image_icon=PhotoImage(file="logo.png")
root.iconphoto(False,image_icon)

si=PhotoImage(file="Untitled.png")
myimage=Label(image=si,bg="#57adff")
myimage.place(x=20,y=20)

textf=tk.Entry(root,justify="center",width=17,font=("poppins",25,"bold"),bg=
"#131414",border=0,fg="white")
textf.place(x=50,y=40)
textf.focus()

sicon=PhotoImage(file="search_icon.png")
myimage_icon=Button(image=sicon,borderwidth=0,
cursor="hand2",bg="#131414",command=getweather)
myimage_icon.place(x=400,y=34)

logoi=PhotoImage(file="logo.png")
logo=Label(image=logoi,bg="#57adff")
logo.place(x=200,y=100)
```

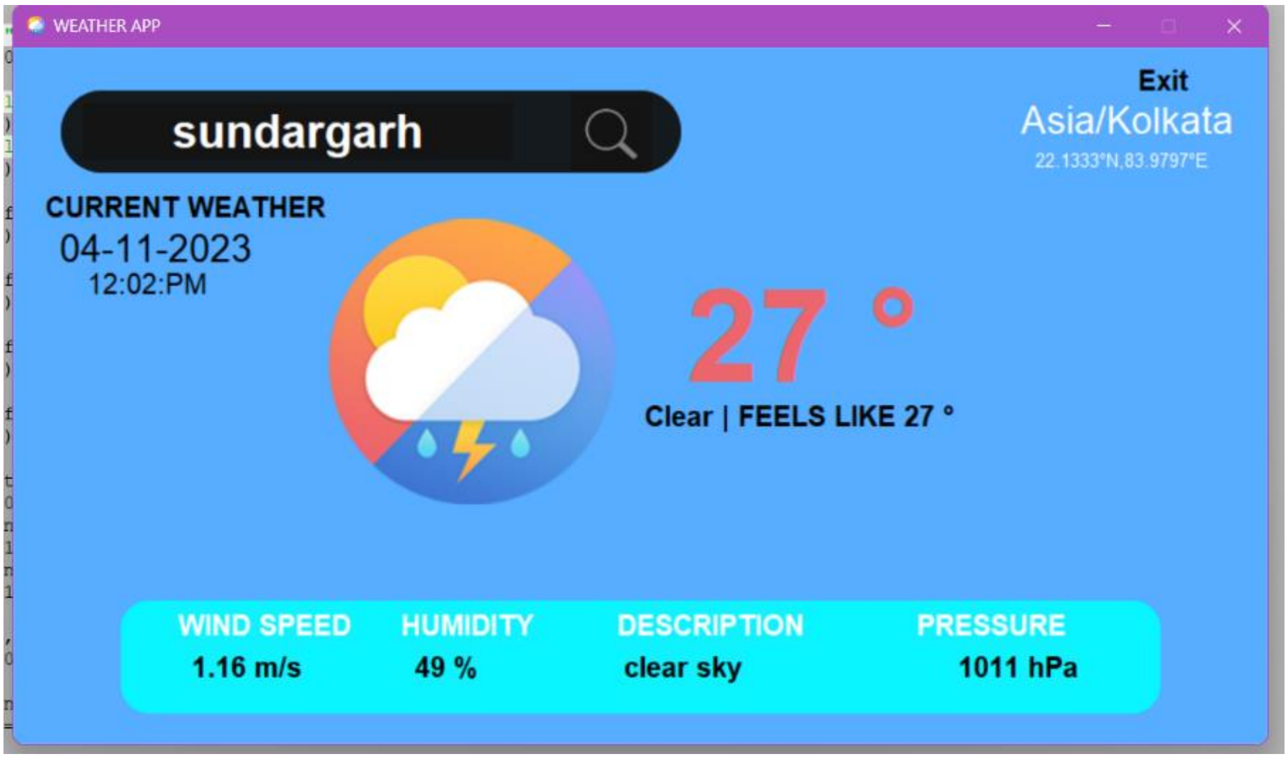
```
fi=PhotoImage(file="box.png")
fmi=Label(image=fi,bg="#57adff")
fmi.pack(padx=5,pady=5,side=BOTTOM)

name=Label(root,font=("arial",15,"bold "),bg="#57adff")
name.place(x=20,y=100)
date1=Label(root,font=("Helvatica",20),bg="#57adff")
date1.place(x=30,y=125)
clock=Label(root,font=("Helvatica",15),bg="#57adff")
clock.place(x=50,y=155)

l1=Label(root ,text="WIND SPEED",
font=("Helvetica",15,"bold"),fg="white",bg="#10F3FF")
l1.place(x=115,y=400)

l2=Label(root,text="HUMIDITY",font=("Helvetica",15,"bold"),fg="white",bg="
#10F3FF")
l2.place(x=275,y=400)

l3=Label(root,text="DESCRIPTION",font=("Helvetica",15,"bold"),fg="white",b
g="#10F3FF")
l3.place(x=430,y=400)

l4=Label(root,text="PRESSURE",font=("Helvetica",15,"bold"),fg="white",bg="
#10F3FF")
l4.place(x=645,y=400)

t=Label(font=("arial",70,"bold"),fg="#ee666d",bg="#57adff")
t.place(x=480,y=150)
c=Label(font=("arial",15,"bold"),bg="#57adff")
c.place(x=450,y=250)

w=Label(text="...",font=("arial",15,"bold"),bg="#10F3FF")
w.place(x=125,y=430)

h=Label(text="...",font=("arial",15,"bold"),bg="#10F3FF")
h.place(x=285,y=430)

d=Label(text="...",font=("arial",15,"bold"),bg="#10F3FF")
```

```
d.place(x=435,y=430)
p=Label(text="...",font=("arial",15,"bold"),bg="#10F3FF")
p.place(x=675,y=430)

timezone=Label(root,font=("Helvetica",20),fg="white",bg="#57adff")
timezone.place(x=720,y=33)

long1=Label(root,font=("Helvetica",10),fg="white",bg="#57adff")
long1.place(x=730,y=70)

exit_button =
tk.Button(root,font=("arial",15,"bold"),bg="#57adff",borderwidth=0,cursor="
hand2", text="Exit", command=exit_application)
exit_button.place(x=800,y=5)

root.mainloop()
#Thankyou
```

HOME PAGE


WEATHER APP
Exit
WIND SPEED
...
HUMIDITY
...
DESCRIPTION
...
PRESSURE
...

OUTPUT



WEATHER APP

Exit

Asia/Kolkata

22.1333°N,83.9797°E

sundargarh

CURRENT WEATHER

04-11-2023

12:02:PM

27 °

Clear | FEELS LIKE 27 °

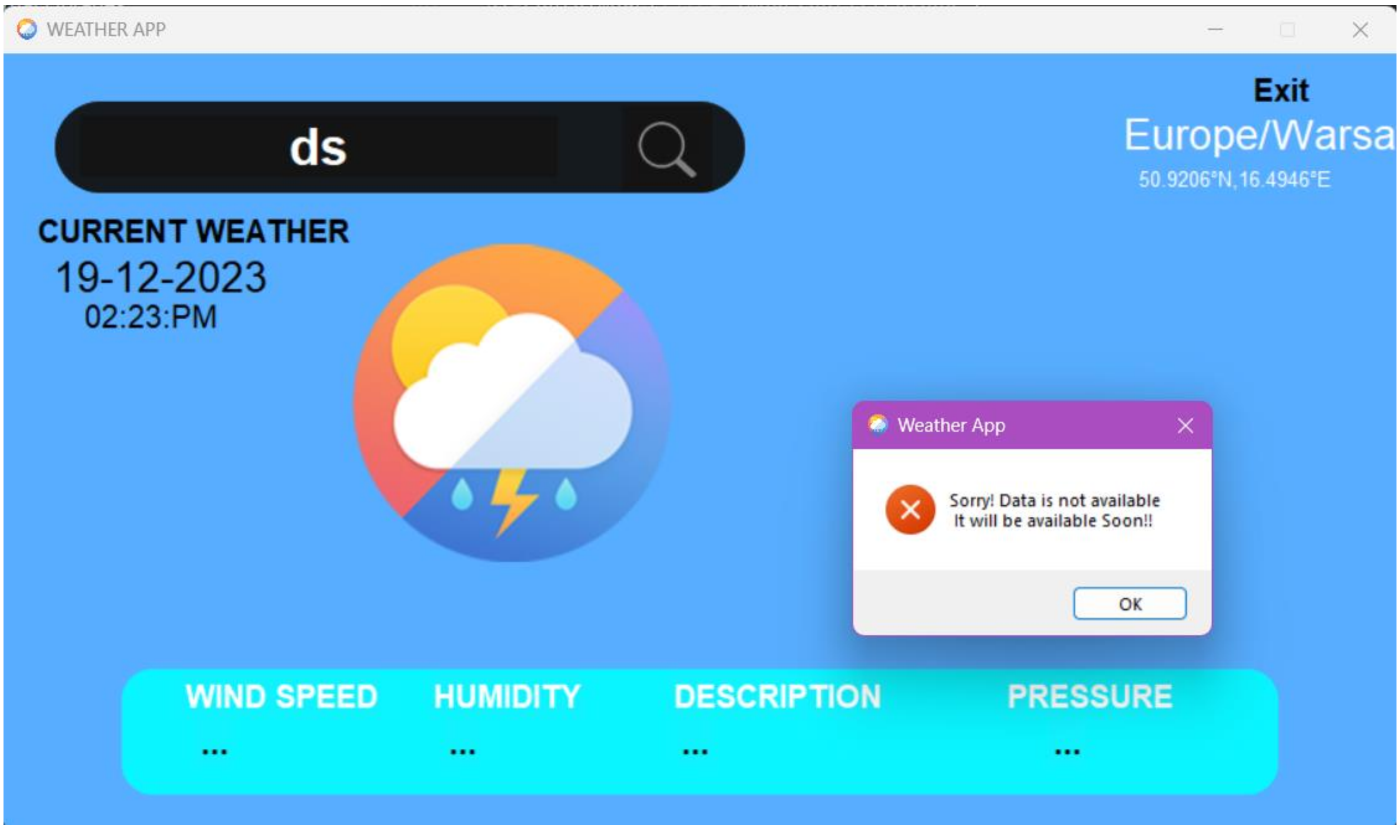
WIND SPEED
1.16 m/s

HUMIDITY
49 %

DESCRIPTION
clear sky

PRESSURE
1011 hPa

IF ERROR OCCUR:-



Sorry! Data is not available
It will be available Soon!!

CONCLUSIONS

- Weather is something everybody deals with, and accurate data of it like what is coming can help users to make inform decisions.

- With weather apps for iOS and Android, people can exactly know when to expect a change in the weather conditions. Weather apps can give urgent alerts too.

- Weather warnings are important because they are used to protect lives and property.

- Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets.

- Temperature forecasts are used by utility companies to estimate demand over coming days.

FUTURE ENHANCEMENTS

Hyper-local Forecasting:

Implement more accurate and hyper-local weather predictions using advanced forecasting models. Include street-level mapping and weather predictions to provide users with highly localized information.

Accessibility Features:

Ensure the app is accessible to users with disabilities by incorporating features such as voice commands, screen readers, and high contrast modes.

Integration with Smart Devices:

Integrate the app with smart home devices, allowing users to receive weather-related notifications on their smart speakers, thermostats, or other connected devices.

LEARNING EXPERIENCE

This project assisted me to gain a practical experience and apply the knowledge assimilated from the previous courses undertook. Putting the knowledge gained earlier and applying different techniques from past courses was interesting and certain concepts, tools and techniques only made sense after seeing their application in a real world scenario.

It was extremely challenging at times but it has been a great and worthwhile learning experience.

There is not at all any doubt that the weather app would be an asset to any company, small or large.

BIBLIOGRAPHY

- *NCERT TEXT BOOK CLASS 12*
- *Sumita Arora class 12 python*
- *Icon.com*
- *Wikipedia*
- *PARENTS , TEACHERS AND FREIND*