WEATHER MONITORING APPLICATION

A Project Report Submitted in Partial Fulfillment of the Requirements

for the degree of B.Tech in Engineering

in

Electronics and Communication Engineering

by:

Name: ARIJIT DAS ADHIKARI; Registration No: 231030121566;

Name: RAKESH JANA; Registration No: 221030110897;

Name: UTPALENDU DAS; Registration No: 231030121577;

Name: SHAKSHAM KUMAR ; Registration No: 231030121572;

Name: SWARAJ MANDAL; Registration No: 231030121576;

Department of Electronics and Communication Engineering

HALDIA INSTITUTE OF TECHNOLOGY

ICARE Complex, Hatiberia

Haldia, Purba Medinipore

Kolkata – 721657

Certificate

This is to certify that that **Arijit Das Adhikari, Rakesh Jana, Utpalendu Das, Shaksham Kumar & Swaraj Mandal** has carried out their project work entitled **"Weather Monitoring Application"** as a part of the curriculum for the B.Tech. in **Electronics & Communication Engineering (ECE)** under Maulana Abul Kalam Azad University of Technology, West Bengal for the year 2023-2024.

This project report is approved by the undersigned only for the purpose for which is submitted. The candidate is entirely responsible for the statements, opinions and conclusions contained herein.

_____

(Surajit Mukherjee)

(Signature of the Mentor)

_____

(Signature of HOD, ECE Dept, Haldia Institute Of Technilogy, with date)

Acknowledgement

We express our special thanks of gratitude to **Surajit Mukherjee** of the Department of Electronics & Communication Engineering (ECE) for her able guidance and valuable support in completing our project.

Heartfelt thanks are also conveyed to all the members of the teaching and non- teaching staff of the Department of ECE for their cordial support and help, whenever needed.

Date:_____

…..……

Arijit Das Adhikari

…..……

Rakesh Jana

…..……

Utpalendu Das

…..……

Shaksham Kumar

…..……

Swaraj Mandal

Contents:

Page No.

1. List of Figures……. 5

2. Abstract……..……. 6

3. Introduction……. 7

4. Literature Review……. 8

5. Methodology …….. 9

    5.1 Theory…… 9

    5.2 Design…… 17

    5.3 Software Used…… 17

    5.4 Program Code…… 18

    5.5 Used Functions..…… 21

    5.6 Weather Comparison & Accuracy Check……. 22

6. Conclusion……..…… 27

7. References…….. 28

List of Figures :

Page no.

- Figure 1. Open Weather Map .…… 13
- Figure 2. Application Interfac….…… 17
- Figure 3. Weather Comparison of Haldia .....…… 22
- Figure 4. Weather Comparison of Goa ……..……. 23
- Figure 5. Weather Comparison of Gujarat ……...……... 24
- Figure 6 Weather Comparison of London …….…….. 25
- Figure 7 Weather Comparison of Darjeeling …….. 26

Abstract :

Weather monitoring applications have become indispensable tools in modern society, offering real-time access to vital meteorological information for individuals, businesses, and organizations across diverse sectors. This project report outlines the development and implementation of a comprehensive weather monitoring app aimed at providing users with accurate and up-to-date weather data. The app utilizes the Open Weather Map API, a widely recognized source of weather information, to fetch current weather conditions, forecasts, and historical data for locations worldwide. Key features of the app include intuitive user interface design, customizable location tracking, detailed weather forecasts,. Through the seamless integration of advanced technologies and user-centric design principles, this weather monitoring app seeks to enhance user experience, facilitate informed decision-making, and promote weather awareness in both personal and professional contexts. This project underscores the importance of leveraging digital innovation to empower individuals and communities with timely and reliable weather information, thereby fostering resilience and preparedness in the face of changing weather patterns and extreme events.

Introduction:

In an era characterized by rapid technological advancements and increasing reliance on digital solutions, weather monitoring applications have emerged as indispensable tools for individuals, businesses, and organizations alike. These applications provide users with real-time access to critical meteorological data, empowering them to make informed decisions, plan activities, and mitigate risks associated with adverse weather conditions. Recognizing the growing importance of such applications, this project report presents the development and implementation of a weather monitoring app designed to offer comprehensive weather information to users across diverse contexts.

In this introduction, we will outline the significance of weather monitoring applications in today's society, discuss the key features and functionalities of our app, and provide an overview of the methodology used in its development. Furthermore, we will highlight the potential benefits of our app in enhancing weather awareness, improving decision-making, and fostering resilience in the face of changing weather patterns and extreme events. Through this project, we aim to demonstrate the value of digital innovation in enhancing weather-related services and empowering individuals and communities to better understand and respond to weather-related challenges.

Literature Review :

4.1 WAETHER MONITORING APPLICATION

The primary objective of this project is to create a user-friendly and feature-rich weather monitoring application that leverages the capabilities of modern technology to deliver accurate and timely weather forecasts, observations, and alerts. The app is built upon the foundation of the Open Weather Map API, a renowned platform for accessing weather data from a wide range of sources, including ground-based weather stations, satellites, and radar systems. By harnessing the vast repository of weather information provided by Open Weather Map, our app aims to provide users with a holistic view of current weather conditions and forecasts for locations worldwide.

The home automation system can be use with various modes.

Methodology :

5.1 Theory :

The theory behind the Weather Monitoring Application is to check and monitor the weather report and weather prediction with great accuracy.

Weather monitoring applications are essential tools for tracking and predicting weather patterns. These applications collect, process, and display real-time weather data from various sources, providing users with accurate and up-to-date information. The development of a weather monitoring application involves integrating data acquisition, processing, storage, and visualization components to deliver a seamless user experience.

5.1.1 Objectives :

The primary objectives of a weather monitoring application include:

- **Data Collection:** Aggregating weather data from reliable sources, such as meteorological stations, satellites, and online APIs.
- **Data Processing:** Analyzing raw data to extract meaningful information and generate weather predictions.
- **Data Storage:** Efficiently storing historical weather data for trend analysis and future reference.
- **User Interface:** Providing an intuitive and interactive interface for users to access weather information.

5.1.2 API (APPLICATION PROGRAMMING INTERFACE)

An API, or Application Programming Interface, serves as an intermediary that enables different software applications to communicate and interact with each other. APIs define the methods and protocols that developers can use to access specific functionalities or data from a software platform or service. Here are some brief details about APIs:

1. **Functionality:** APIs expose certain functionalities or data from a software system, allowing developers to access and utilize them in their own applications without needing to understand the internal workings of the system.

2. **Standardization:** APIs often adhere to standardized protocols and formats, such as REST (Representational State Transfer) or SOAP (Simple Object Access Protocol), which facilitate interoperability and ease of integration across different systems and programming languages.

3. **Types of APIs:** There are various types of APIs, including:

    a. **Web APIs:** APIs that are accessed over the internet using HTTP requests. These are commonly used for accessing web services and online platforms.

    b. **Library APIs:** APIs provided by software libraries or frameworks that developers can use to interact with the library's functionality.

    c. **Operating System APIs:** APIs provided by operating systems that allow applications to interact with system resources such as files,

hardware devices, and network interfaces.

4. **Authentication and Authorization:** APIs may require authentication and authorization mechanisms to ensure that only authorized users or applications can access protected resources or perform certain actions.

5. **Usage:** Developers integrate APIs into their applications by making API calls, which typically involve sending HTTP requests with specific parameters to the API endpoint and receiving responses containing the requested data or indicating the success or failure of the operation.

6. **Documentation:** API providers often offer documentation that describes the available endpoints, parameters, request formats, response formats, and authentication methods, which helps developers understand how to use the API effectively.

Overall, APIs play a crucial role in enabling interoperability, integration, and extensibility in software development, allowing developers to leverage existing functionalities and services to build powerful and innovative applications.

5.1.3 OPEN WEATHER MAP API

The Open Weather Map API is a widely-used platform that provides developers with access to a wealth of weather data and related services. Here are some key details about the Open Weather Map API:

5.1.3.1 Weather Data: The Open Weather Map API offers access to a wide range of weather data, including current weather conditions, forecasts, historical weather data, and weather maps. This data covers various parameters such as temperature,

humidity, wind speed and direction, atmospheric pressure, precipitation, and more.

5.1.3.2 Global Coverage: Open Weather Map provides weather data for locations worldwide, making it a valuable resource for developers building applications with global reach.

5.1.3.3 API Endpoints: The API offers different endpoints for accessing specific types of weather data. These endpoints include:

- Current weather data
- Hourly and daily forecasts
- Historical weather data
- Weather maps (e.g., radar, satellite)

5.1.3.4 Customization: The Open Weather Map API allows developers to customize their requests by specifying parameters such as location coordinates, units of measurement (e.g., metric or imperial), language, and data format (e.g., JSON or XML).

5.1.3.5 Free and Paid Plans: Open Weather Map offers both free and paid plans for accessing their API. The free plan typically comes with usage limits and may have restricted access to certain features, while paid plans offer higher usage limits, premium data, and additional features.

5.1.3.6 Documentation: The API is well-documented, providing developers with comprehensive information on available endpoints, request parameters, response formats, authentication methods, usage limits, and examples. This documentation is essential for developers to understand how to effectively use the API in their applications.

5.1.3.7 Community Support: Open Weather Map has a large community of developers who use the API and contribute to its ecosystem. This community support can be valuable for developers seeking assistance, sharing best practices, and collaborating on projects involving weather data.

Overall, the Open Weather Map API is a powerful tool for developers looking to integrate weather data into their applications, whether for personal use, business applications, or research purposes. Its extensive coverage, customization options, and documentation make it a popular choice among developers worldwide.

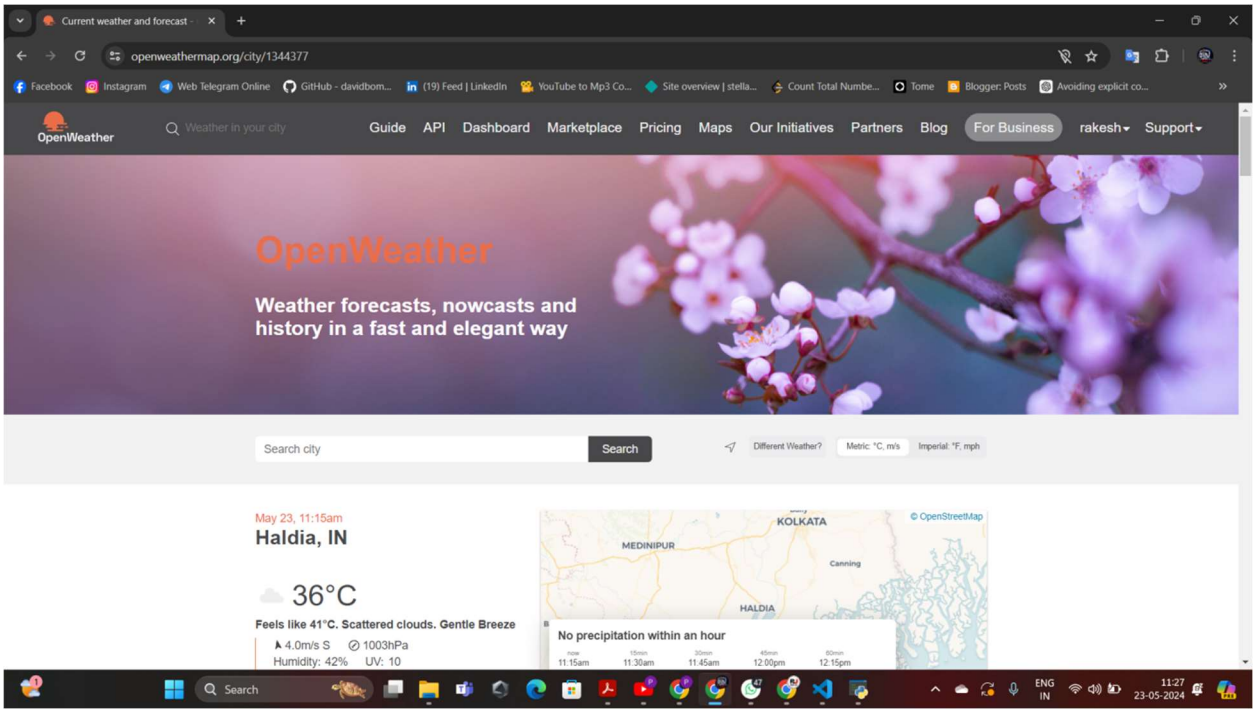


Figure 1 : Open Weather Map

5.1.4 PYTHON PROGRAMMING LANGUAGE

Overall, Overall, the Open Weather Map API is a powerful tool for developers looking to integrate weather data into their applications, whether for personal use, business applications, or research purposes. Its extensive coverage, customization options, and documentation make it a popular choice among developers worldwide.

5.1.4.1 Key Features of Python:

1. **Readability and Simplicity:** Python's syntax is clean and straightforward, which makes it easy to learn and write. Its use of indentation for block delimiters is a distinctive feature that enhances readability.

2. **Interpreted Language:** Python is an interpreted language, meaning code is executed line-by-line, which facilitates rapid development and testing.

3. **Dynamically Typed:** Variables in Python do not require explicit declaration of data types, allowing for more flexible and concise code.

4. **Versatility:** Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. This versatility makes it suitable for a wide range of applications, from web development to data analysis and artificial intelligence.

5. **Extensive Standard Library:** Python comes with a comprehensive standard library that includes modules and packages for common tasks, such as file I/O, system calls, and data manipulation.

6. **Large Ecosystem and Community:** Python boasts a vast ecosystem of third-party packages available through the Python Package Index (PyPI). Its large and active community contributes to a wealth of resources, including tutorials, documentation, and forums.

7. **Cross-Platform Compatibility:** Python is available on multiple operating systems, including Windows, mac OS, and various distributions of Linux, allowing developers to create cross-platform applications with ease.

8. **Integration Capabilities:** Python can easily integrate with other languages and technologies, such as C, C++, Java, and .NET, making it a flexible choice for various projects.

5.1.4.2 Common Uses of Python:

- Web Development: Frameworks like Django and Flask make it easy to build robust web applications.
- Data Science and Machine Learning: Libraries such as NumPy, pandas, and scikit-learn are widely used for data analysis and machine learning tasks.
- Automation and Scripting: Python's simplicity and ease of use make it ideal for automating repetitive tasks and writing scripts.
- Scientific Computing: Tools like SciPy and Matplotlib support scientific and engineering calculations.
- Artificial Intelligence: Python is a preferred language for AI development, with libraries like TensorFlow and PyTorch.

Python's combination of simplicity, versatility, and a supportive community makes it a powerful tool for developers across many domains.

5.2 DESIGN :

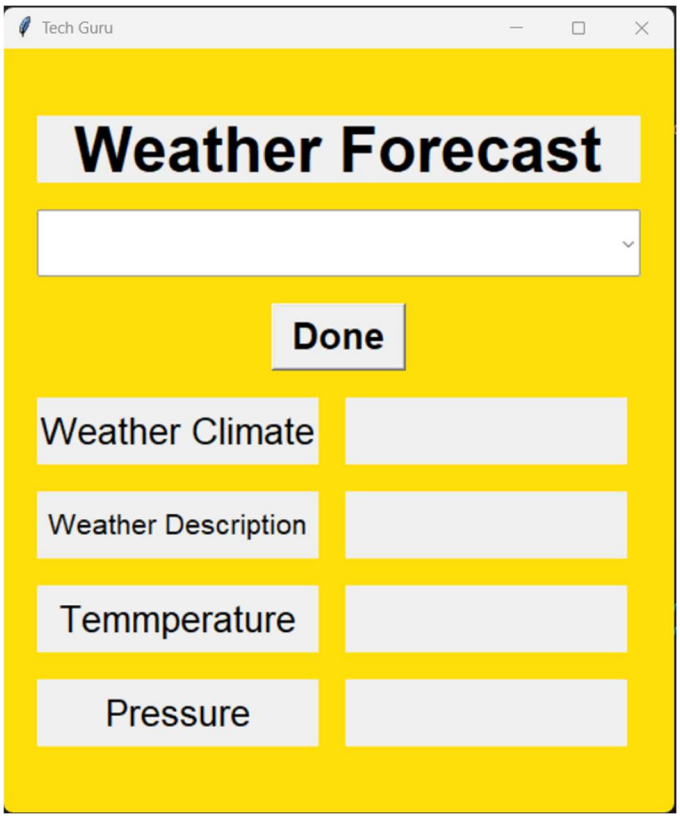


Figure 2 : Application Interface

5.3 Software used :

7.3.1 Text Editor : Microsoft Visual Studio Code

7.3.2 Programming Language : Python Programming Language

7.3.3 API Used : Open Weather Map API

5.4 Program Code :

```
from tkinter import *
from tkinter import ttk

#import weather api ---------------------->
import requests
def data_get():
    city=city_name.get()

data=requests.get("https://api.openweathermap.org/data/2.5/weather?q="+city+"&appid=74adf478e3ad35
37384096fcfffcdf44").json()
    w_label1.config(text=data["weather"][0]["main"])


    wb_label1.config(text=data["weather"][0]["description"])


    temp_label1.config(text=str((int(data["main"]["temp"]-273)))+u"°C")


    pre_label1.config(text=data["main"]["pressure"])




win = Tk()
win.title("Tech Guru")
win.config(bg="#ffdf09")
win.geometry("500x570")

name_label=Label(win,text="Weather Forecast",font=("Time New Roman",35,"bold"))
name_label.place(x=25,y=50,height=50,width=450)
```

```
city_name=StringVar()

list_name=["Andhra Pradesh","Arunachal Pradesh
","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana","Himachal Pradesh","Jammu and
Kashmir","Jharkhand","Karnataka","Kerala","Madhya
Pradesh","Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab","Rajasthan",
"Sikkim","Tamil Nadu","Telangana","Tripura","Uttar Pradesh","Uttarakhand","West Bengal","Andaman
and Nicobar Islands","Chandigarh","Dadra and Nagar Haveli","Daman and
Diu","Lakshadweep","National Capital Territory of Delhi","Puducherry"]

com=ttk.Combobox(win,text="Weather Forecast",values=list_name,font=("Time New
Roman",20,"bold"),textvariable=city_name)

com.place(x=25,y=120,height=50,width=450)




#weather_climate---------->

w_label=Label(win,text="Weather Climate",font=("Time New Roman",20,))

w_label.place(x=25,y=260,height=50,width=210)


w_label1=Label(win,text="",font=("Time New Roman",20,))

w_label1.place(x=255,y=260,height=50,width=210)


#weather_description---------->

wb_label=Label(win,text="Weather Description",font=("Time New Roman",16,))

wb_label.place(x=25,y=330,height=50,width=210)


wb_label1=Label(win,text="",font=("Time New Roman",20,))

wb_label1.place(x=255,y=330,height=50,width=210)



#temperature---------->

temp_label=Label(win,text="Temmperature",font=("Time New Roman",20,))
```

```
temp_label.place(x=25,y=400,height=50,width=210)

temp_label1=Label(win,text="",font=("Time New Roman",20,))
temp_label1.place(x=255,y=400,height=50,width=210)



#pressure---------------------->
pre_label=Label(win,text="Pressure",font=("Time New Roman",20,))
pre_label.place(x=25,y=470,height=50,width=210)

pre_label1=Label(win,text="",font=("Time New Roman",20,))
pre_label1.place(x=255,y=470,height=50,width=210)



#done button---------------->
done_button=Button(win,text="Done",font=("Time New Roman",20,"bold"),command=data_get )
done_button.place(y=190,height=50,width=100,x=200)

win.mainloop()
```

5.5 Used Functions :

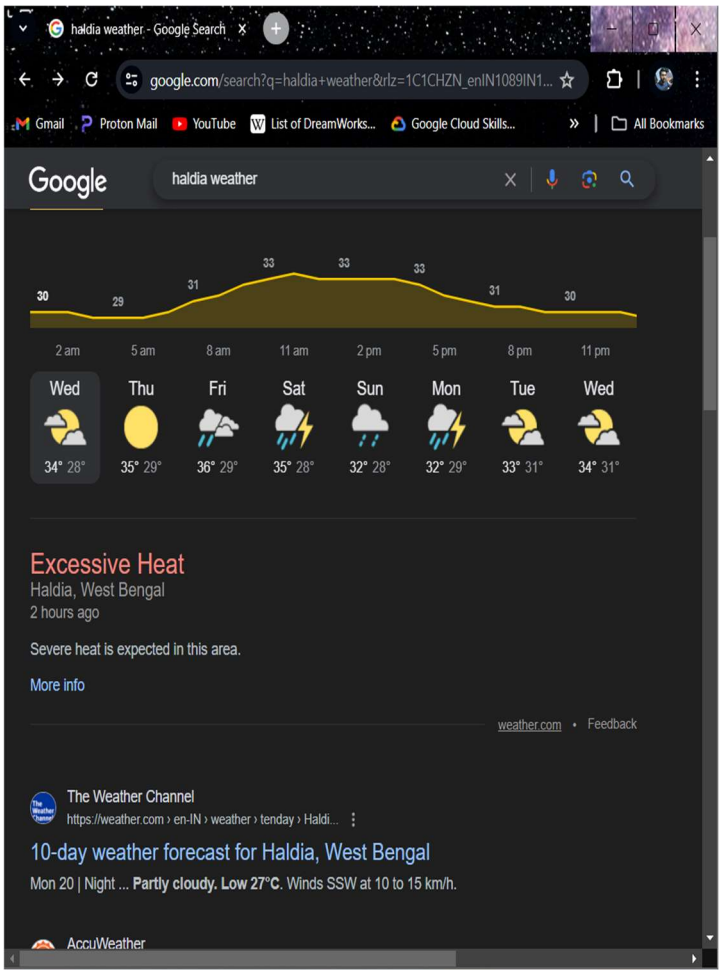
- Tkinter : Tkinter is the standard GUI (Graphical User Interface) library for Python, which provides a fast and easy way to create desktop applications.

- **Ttk :** ‘Ttk’ is a module in Python's Tkinter library that stands for "Tk Themed Widgets". It provides a set of widgets that are designed to give a more modern look and feel to your Tkinter applications, with the added benefit of being customizable through themes. ‘Ttk’ widgets often look more polished and can be styled to match different platforms more closely, making your application look more native.

- **Requests :** The requests library in Python is a powerful and user-friendly HTTP library that allows you to send HTTP requests and handle responses in a straightforward manner. It abstracts the complexities of making HTTP requests behind a simple API, making it easy to interact with web services and APIs.

- **Win.mainloop() :** In Python, specifically when using the Tkinter library to create graphical user interfaces (GUIs), ‘win.mainloop()’ plays a crucial role. Here, ‘win’ typically represents an instance of ‘tk.Tk()’ or a window created in the Tkinter framework. The ‘mainloop()’ method is responsible for starting the Tkinter event loop, which is necessary to keep the application running and responsive to user interactions.

5.6 Weather Comparison & Accuracy Check :



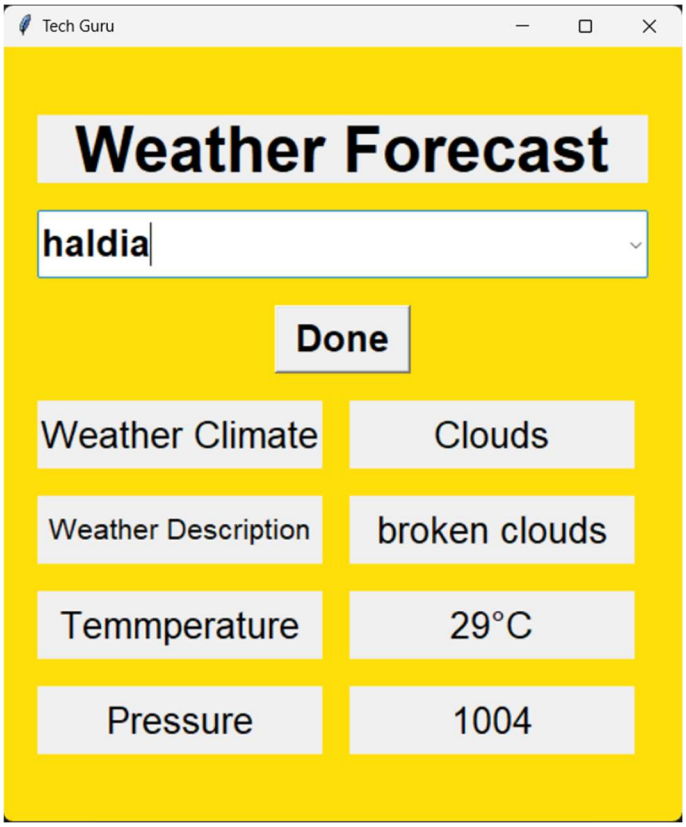


Figure 3 : Weather Comparison of Haldia

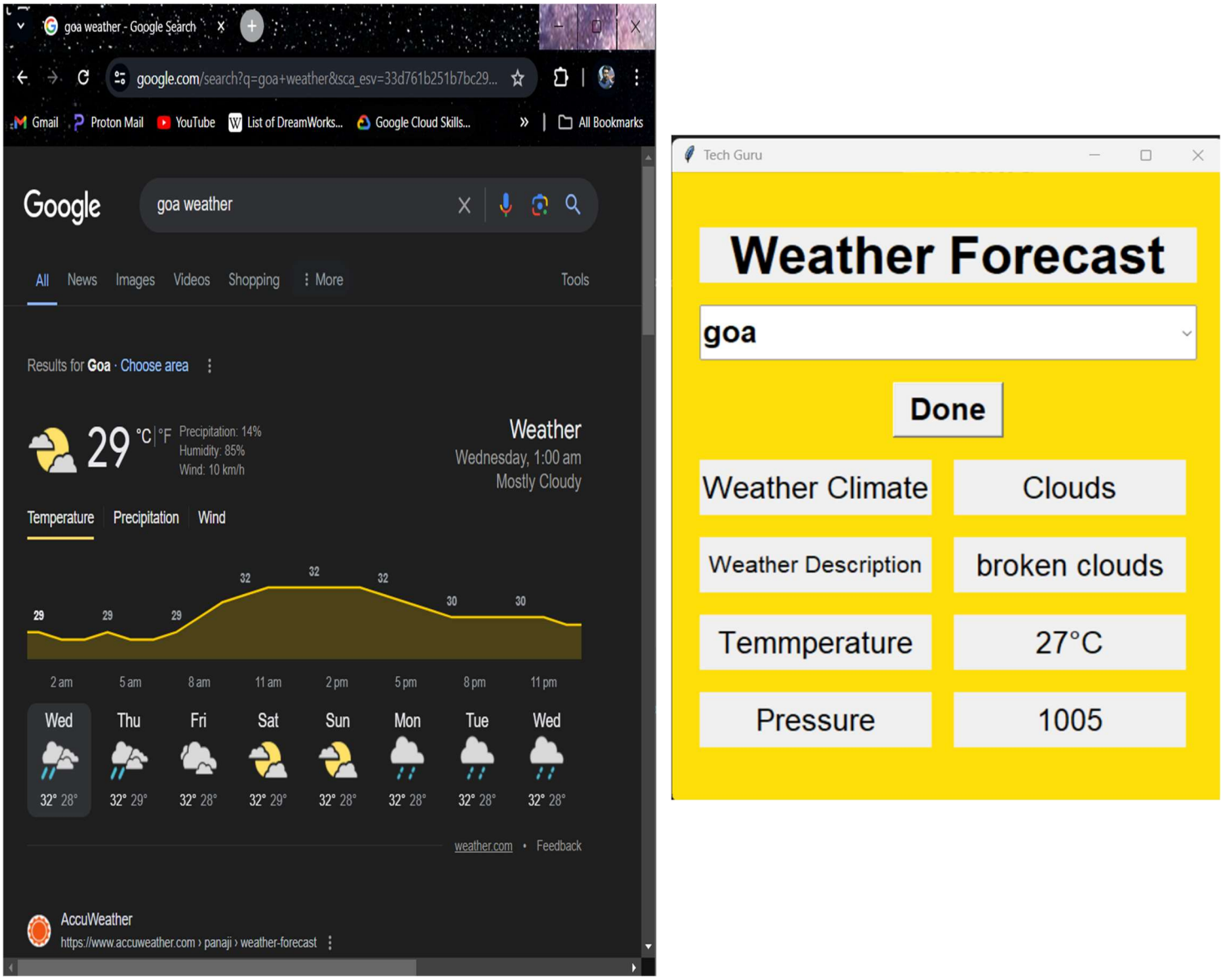
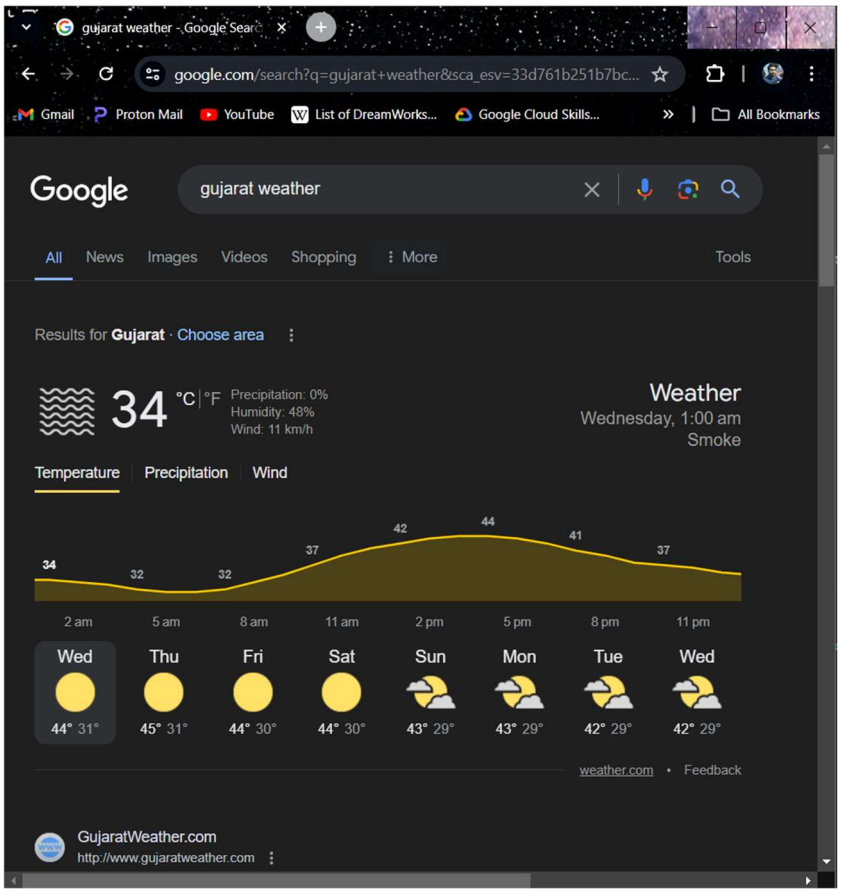


Figure 4 : Weather Comparison of Goa



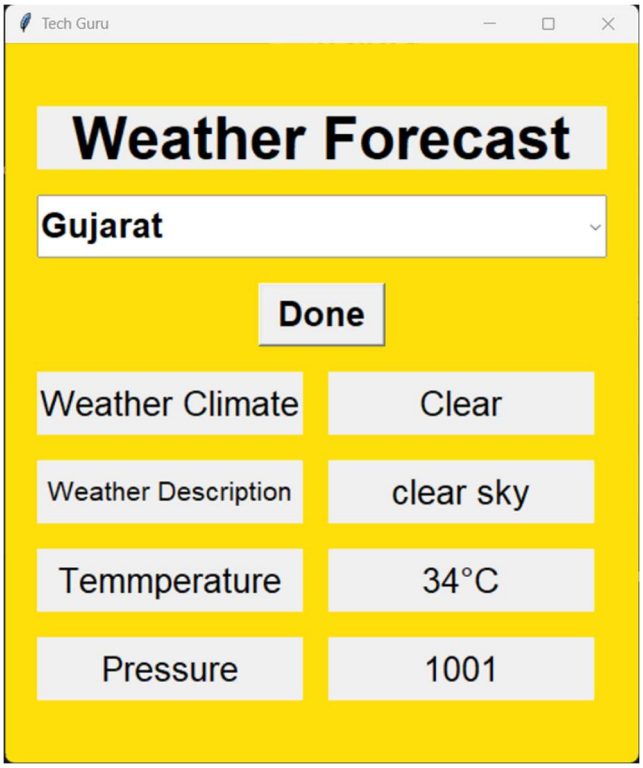


Figure 5 : Weather Comparison of Gujarat

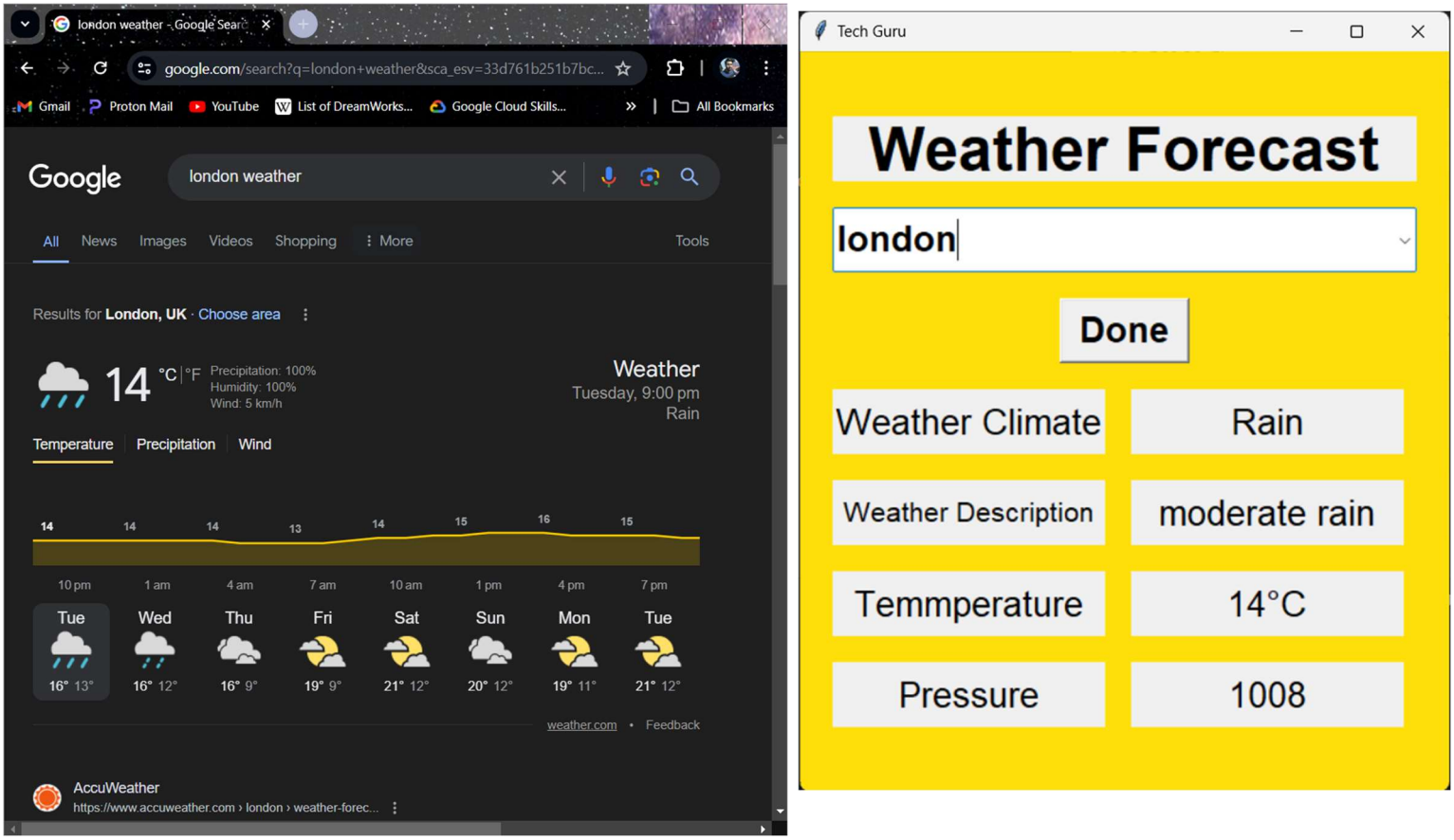
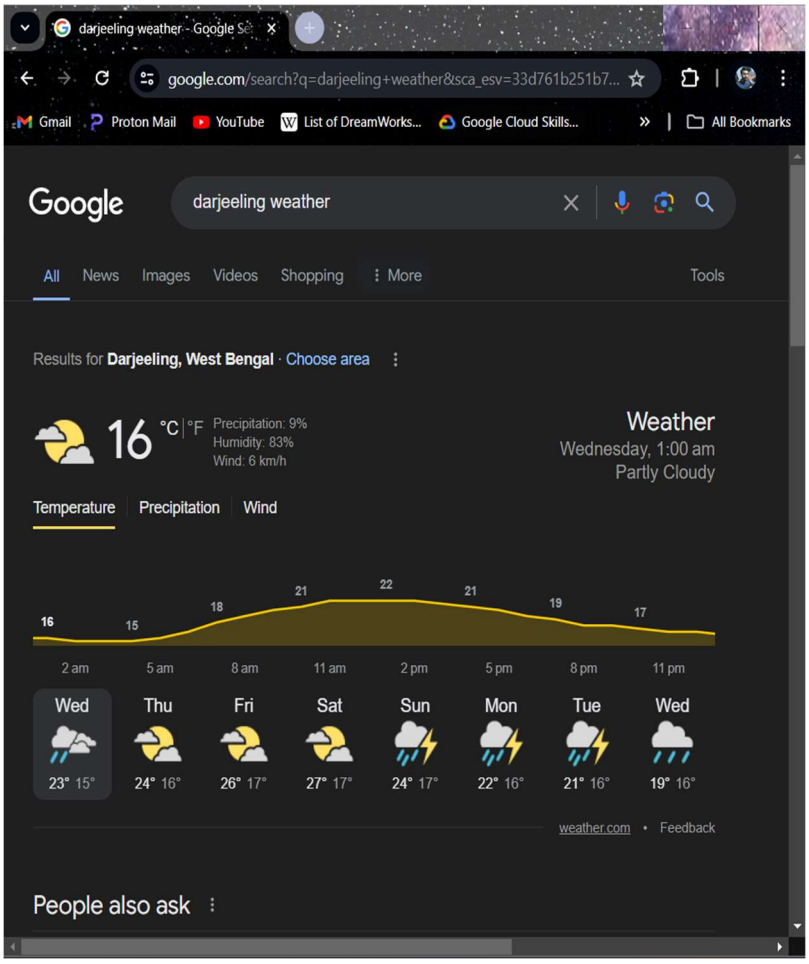


Figure 6 : Weather Comparison of London



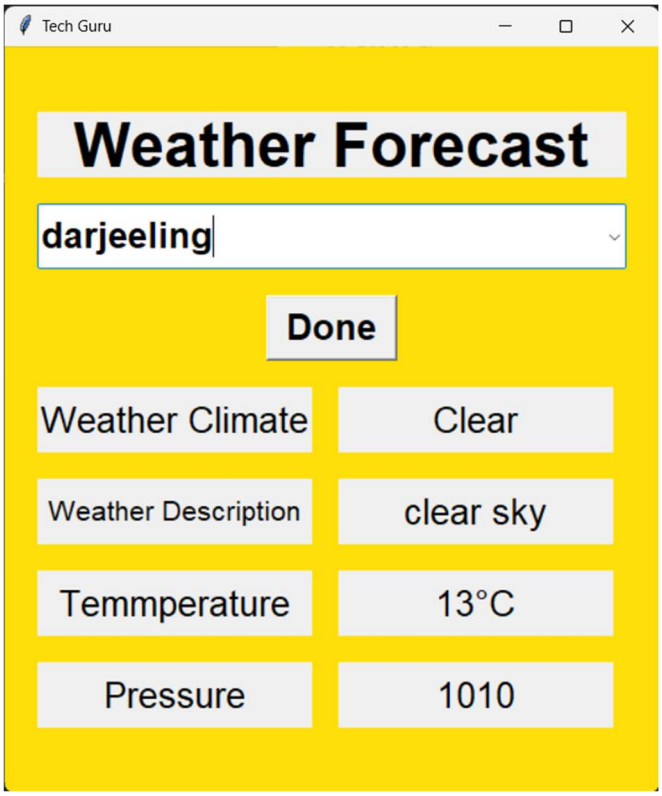


Figure 7 : Weather Comparison of Darjeeling

CONCLUSION :

The development and deployment of the weather monitoring application have been a comprehensive project integrating various aspects of software development, data analysis, and user interface design. This application, designed to provide real-time weather updates and forecasts, has successfully met its objectives through the following key outcomes:

Real-Time Data Integration:

- The application effectively utilizes APIs from reliable weather data providers to fetch real-time weather information. This ensures that users have access to the most current weather conditions and forecasts.

User-Friendly Interface:

- A significant focus was placed on the user experience. The interface is intuitive and easy to navigate, catering to users of all ages and technical backgrounds. Clear display of weather data, including temperature, humidity, wind speed, and precipitation, ensures that users can quickly comprehend the information presented.

Data Visualization:

- Advanced data visualization techniques have been employed to present weather trends and forecasts. Graphs and charts provide a visual representation of weather patterns, enhancing the user's ability to interpret and understand complex data.

References:

Citations from Internet :

[1] https://youtu.be/G-FBEDM7b3Y?si=c25SS4HCgIr60Ppb

[2] https://openweathermap.org/