



## **FINAL PROJECT PRESENTATION**

Weather data analysis using Excel and Python

### **SUBMITTED BY:**

- Noor-ul-Huda D/o Sohail Ahmed (Roll no:478460)

### **COURSE NAME:**

CIT (Computer Information Technology)

### **SUBMITTED TO:**

Sir Fahad bin Ashfaq

## **1.INTRODUCTION**

Weather plays an important role in daily life and industries such as agriculture and transportation. Accurate weather data helps in better planning and decision-making. In this project, weather data is collected and organized using Microsoft Excel.

The data is then analyzed using Python with libraries like Pandas and Matplotlib. This integration helps in performing calculations and generating graphs to understand weather trends effectively.

## **2.OBJECTIVES**

- To obtain historical weather data from the Meteostat website.
- To store and organize the extracted data in Microsoft Excel.
- To read and analyze the Excel data using Python.
- To interpret weather trends through basic data analysis and visualization.

### **3. TOOLS & TECHNOLOGIES**

- **Meteostat Website** – Used to extract and export historical weather data.
- **Microsoft Excel** – Used to store and organize the collected weather data.
- **Python (Pandas & Openpyxl Libraries)** – Used to read, write, and analyze the Excel data.
- **VS Code** – Used as the development environment to write and execute the Python code.
- **GitHub** – Used to upload and manage the project files online.

### **4. METHODOLOGY**

#### **1. Data Collection:**

Historical weather data for the selected city and time period was obtained from the **Meteostat website**.

#### **2. Data Storage:**

The extracted data was downloaded and organized in **Microsoft Excel** to ensure proper structure and easy access.

#### **3. Data Import in Python:**

The Excel file was imported into **Python** using **Pandas** and **Openpyxl** libraries.

## 4. Data Analysis:

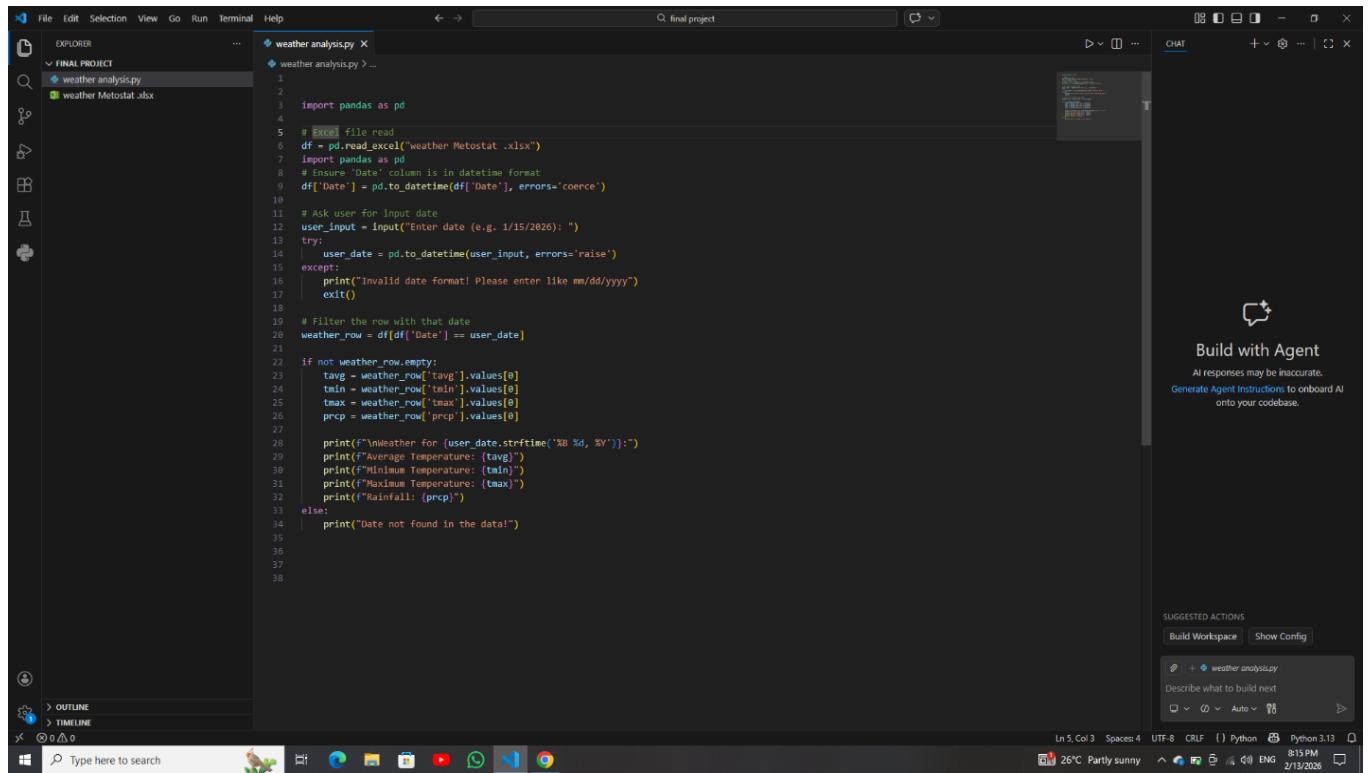
The weather data was analyzed to calculate basic statistics such as average, maximum, and minimum values.

## 5. Results Interpretation:

The analyzed data were interpreted to understand weather trends for the selected period.

## 5. Implementation

### Inserting python code in VS CODE:



A screenshot of the Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "weather analysis.py". The Explorer sidebar on the left lists a "FINAL PROJECT" folder containing "weather analysis.py" and "weather Metostat.xlsx". The main editor area displays the following Python code:

```
1 import pandas as pd
2
3 # Excel file read
4 df = pd.read_excel("weather Metostat.xlsx")
5 import pandas as pd
6 # Ensure 'Date' column is in datetime format
7 df["Date"] = pd.to_datetime(df["Date"], errors='coerce')
8
9 # Ask user for input date
10 user_input = input("Enter date (e.g. 1/15/2026): ")
11 try:
12     user_date = pd.to_datetime(user_input, errors='raise')
13 except:
14     print("Invalid date format! Please enter like mm/dd/yyyy")
15     exit()
16
17 # Filter the row with that date
18 weather_row = df[df['Date'] == user_date]
19
20 if not weather_row.empty:
21     tavg = weather_row['tavg'].values[0]
22     tmin = weather_row['tmin'].values[0]
23     tmax = weather_row['tmax'].values[0]
24     precip = weather_row['prcp'].values[0]
25
26     print(f"\nWeather for {user_date.strftime('%B %d, %Y')}:")
27     print(f"Average Temperature: ({tavg})")
28     print(f"Minimum Temperature: ({tmin})")
29     print(f"Maximum Temperature: ({tmax})")
30     print(f"Rainfall: ({precip})")
31 else:
32     print("Date not found in the data!")
33
34
35
36
37
38
```

The bottom status bar shows "Ln 5, Col 3 Spaces: 4 UFT-8 Python 3.13" and the system tray indicates "26°C Partly sunny 8:15 PM 2/13/2026". A "Build with Agent" panel is visible on the right side of the interface.

## 6. Short Explanation about code

- 1. Import pandas** – I import pandas library to work with Excel files and tables.
- 2. Load Excel file** – The file weather\_data.xlsx is read into a table called df.
- 3. Convert Date column** – The Date column is converted to proper datetime format so Python can understand it.
- 4. Ask user for input** – The program asks the user to type a date in mm/dd/yyyy format.
- 5. Check date format** – The input is converted to datetime; if it's wrong, an error message is shown and the program exits.
- 6. Filter the row** – The program searches the Excel data for the row that matches the user's date.
- 7. Check if date exists** – If a row is found, it proceeds; otherwise, it prints “Date not found.”
- 8. Get weather values** – From the matching row, it gets average temperature (tavg), min (tmin), max (tmax), and rainfall (prcp).
- 9. Print results** – The program prints the weather info neatly for the user.

This covers the main flow of my script in simple points.

## **7. OUTPUT & RESULT**

Python program output is below:

```
weather analysis.py
weather analysis.py > ...
1
2
3 import pandas as pd
4
5 # Excel file read
6 df = pd.read_excel("weather Metostat .xlsx")
7 import pandas as pd
8 # Ensure 'Date' column is in datetime format
9 df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
10
11 # Ask user for input date
12 user_input = input("Enter date (e.g. 1/15/2026): ")
13 try:
14     user_date = pd.to_datetime(user_input, errors='raise')
15 except:
16     print("Invalid date format! Please enter like mm/dd/yyyy")
17     exit()
18
19 # Filter the row with that date
20 weather_row = df[df['Date'] == user_date]
21
22 if not weather_row.empty:
23     tavg = weather_row['tavg'].values[0]
24     tmin = weather_row['tmin'].values[0]
25     tmax = weather_row['tmax'].values[0]
26     prcp = weather_row['prcp'].values[0]
27
28     print("Weather for {date.strftime('%B %d, %Y')}:")
29     print(f"Average Temperature: {tavg}")
30     print(f"Minimum Temperature: {tmin}")
31     print(f"Maximum Temperature: {tmax}")
32     print(f"Rainfall: {prcp}")
33 else:
34     print("Date not found in the data!")
35
```

PS C:\Users\User\Desktop\final project> & C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/User/Desktop/final project/weather analysis.py"  
Enter date (e.g. 1/15/2026): 1/26/2026  
Weather for January 26, 2026:  
Average Temperature: 12.6  
Minimum Temperature: 12.6  
Maximum Temperature: 22.0  
Rainfall: 0.9  
PS C:\Users\User\Desktop\final project>

## 8.CONCLUSION

This program allows the user to retrieve weather information like average, minimum, and maximum temperature, and rainfall for any date from an Excel file. It ensures correct date format and handles cases when the date is not found.

یہ پروگرام صارف کو کسی بھی تاریخ کے لیے اوسط، کم از کم اور زیادہ سے زیادہ درجہ حرارت اور بارش کی معلومات حاصل کرنے دیتا ہے۔ یہ درست تاریخ کے فارمیٹ کو یقینی بناتا ہے اور اگر تاریخ موجود نہ ہو تو مناسب پیغام دکھاتا



