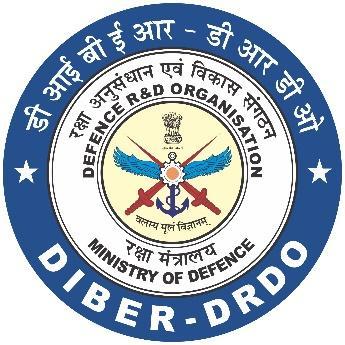
**SUMMER TRAINING REPORT**

on

**“Smart Scraper: Cross-Site Weather Data Collection and Trend Analysis”**



Project Report

UNDER THE GUIDANCE of

**SH. PIYUSH JOSHI, SCIENTIST ‘F’**

**DIBER (A Cell of DIPAS), DRDO**

SUBMITTED BY

**SHIVANI AGARWAL**

BACHELOR OF TECHNOLOGY, F INAL YEAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA DEEMED TO BE UNIVERSITY

DEHRADUN, UTTARAKHAND

**Certificate**

This is to certify that **Shivani Agarwal**, a student of **Graphic Era Deemed to be University** has successfully completed the project titled **“Smart Scraper: Cross-Site Data Collection & Trends Analysis”** as a part of the partial fulfillment of the requirements for the **Bachelor of Technology in Computer Science**.

The project involves extracting weather and data from online sources, processing and visualizing the collected data, and exporting the structured outputs into separate CSV files for each source.

The work carried out in this project is completed under the guidance of **Sh. Piyush Joshi, Sc. ‘F’, DIBER (A Cell of DIPAS).**

**(Sh. Piyush Joshi)**

**Sc. ‘F’, DRDO, (DIBER)**

**ACKNOWLEDGEMENT**

I express my heartfelt gratitude to **Sh. Devakanta Pahad Singh, Director, DIPAS**, for granting me the opportunity to work on this project at DIBER and for providing the necessary approval and facilities to carry out this work.

I would like to express my profound gratitude and indebtedness to my guide, **Sh. Piyush Joshi, Sc. ‘F’,DIBER (A Cell of DIPAS)** for his constant support and guidance, without which I would not have been able to attempt this project.

I thank him for being a constant help throughout the project and encouraging me with his words and whose regular useful advice, time, patience and understanding helped me a lot. His suggestions and instructions have served as the major contribution towards the completion of the project.

I would also like to thank **Mr. Dhirendra, SRF, DIBER (A Cell of DIPAS)** for also providing his invaluable guidance and technical assistance. His insightful feedback and encouragement were extremely helpful throughout my project.

**Shivani Agarwal**

Bachelor of technology, Final Year

Dept. of Computer Science & Engineering

Graphic Era Deemed to be University, Dehradun

**Abstract**

This project focuses on the extraction, visualization, and comparative analysis of weather data sourced from different platforms. The objective is to assess the consistency and reliability of weather parameters - **temperature, humidity, and wind speed** reported by each source.

Data scraping techniques were employed to programmatically extract real-time and historical weather data from both websites. The extracted data was then stored in structured **CSV files** for each source independently. Following this, the data was visualized using various graphical tools to identify trends, anomalies, and correlations across the selected parameters.

Finally, a comparative analysis was conducted between the two datasets to evaluate discrepancies and agreement levels in reported values. The results offer valuable insights into data accuracy, reporting standards, and usability of open-source meteorological data for research and decision-making purposes.

This study not only highlights the significance of data validation across platforms but also contributes towards enhancing the reliability of weather-based predictions and climate research.

**Table of Content**

1. **Certificate**
2. **Acknowledgement**
3. **Abstract**
4. **Introduction**
   1. Project Overview
   2. Project Scope
   3. Problem statement
   4. Objective
5. **Review of Literature**
   1. Web Scraping: State-of-the-Art and Areas of Application
   2. Web Scraping Techniques to Collect Weather Data in South Sumatera
   3. A Comparative Analysis of Data Mining Methods for Weather Prediction
   4. A Web Scraping Framework for Meteorological Big Data
   5. Weather Report on Metropolitan Indian Cities via Web Scraping
   6. Data Analysis by Web Scraping using Python
6. **Methodology**
   1. Study Area
   2. Data Collection and Sources
   3. Input Features and Target Variable
   4. Data Preprocessing
   5. Program Implementation
7. **Result and Discussion**
   1. Results an Outcomes
   2. Visualization
   3. Comparison Analysis
8. **Summary and Conclusion** 
   1. Summary
   2. Conclusion

**6. References**

**Chapter 1: Introduction**

**1.1 Project Overview**

With online platforms offering meteorological information, it becomes important to understand the consistency and reliability of their data. This project focuses on collecting and analyzing weather data from two major sources: **Meteostat, Wunderground, Visual Crossing**, **Visual Crossing** specifically targeting parameters such as **temperature**, **humidity**, and **wind speed**.

Data is extracted using web scraping techniques and stored in structured **CSV files** for each source independently. This structured data is then visualized using tools like **Python**, **Pandas**, and **Matplotlib** to identify patterns and assess temporal behavior.

A comparison is performed to examine similarities and differences between the two datasets, helping to highlight discrepancies and reporting variations.

**1.2 Project Scope**

This project focuses on collecting, processing, and analyzing weather data from two publicly available platforms, **Meteostat, Wunderground, Visual Crossing**. The scope includes the following key aspects:

* **Data Extraction:** Automated scraping of weather parameters temperature, humidity, and wind speed from both websites for a selected location and defined time period.
* **Data Storage:** The extracted data is cleaned and converted into structured **CSV files** separately for each source, ensuring easy accessibility and further processing.
* **Data Visualization:** Creating visual representations such as line charts and trend graphs to observe temporal patterns and variations within each dataset.
* **Comparative Analysis:** Evaluating the consistency and differences between the datasets from the two platforms to assess data reliability and reporting standards.

The scope emphasizes practical data and highlights how extracted data supports forecasting applications, providing insights for researchers and decision-makers relying on open-source meteorological information.

**1.3 Problem Statement**

Accurate and consistent weather data is essential for effective analysis and forecasting. However, weather information obtained from different online platforms often varies due to differences in data collection methods, sensor calibrations, and update intervals. These inconsistencies create challenges when users attempt to rely on data for research or predictive modeling.

Meteostat, Wunderground, Visual Crossing are used weather data sources, but their reported values for key parameters such as temperature, humidity, and wind speed sometimes show discrepancies. Additionally, the data formats and availability differ between these platforms, complicating efforts to integrate and analyze the information effectively.

This project addresses these challenges by scraping weather data from both sources, converting the extracted data into standardized CSV files. The data is then visualized and compared to identify inconsistencies and evaluate the reliability of each platform.

By conducting this comparative analysis, the project aims to highlight differences in data quality and provide a foundation for selecting trustworthy data sources, ultimately supporting more accurate and reliable weather forecasting.

**1.3 ObjectiveObjective**

The primary objective of this project is to collect, process, and analyze weather data from major online platforms, **Meteostat, Wunderground, Visual Crossing**, focusing on key parameters such as temperature, humidity, and wind speed. By extracting this data through web scraping and organizing it into structured CSV files, the project aims to enable effective visualization and comparative analysis of the datasets.

This comparison seeks to identify discrepancies, assess the consistency of reported values. Additionally, the project supports the use of this processed data for weather forecasting applications, contributing to more accurate and data-driven predictive models.

Overall, the project strives to improve understanding of open-source weather data quality and to provide a foundation for informed decision-making in weather-related research and operational fields.

**1.4 Applications**

* **Weather Forecasting:** The processed and validated weather data can be used to improve forecasting models by providing reliable input parameters such as temperature, humidity, and wind speed.
* **Data Validation:** Researchers and analysts can use the comparative insights from sources to validate and cross-check weather information before using it for scientific studies or operational purposes.
* **Climate Studies:** Historical weather data collected and cleaned through this project can support climate trend analysis and environmental monitoring.
* **Decision Support Systems:** Accurate and consistent weather data aids in developing systems that assist in planning and operational decisions based on weather conditions.

**Chapter 2: Review of Literature**

**2.1 “ Web Scraping: State-of-the-Art and Areas of Application”**

This paper reviews the **current state of web scraping technologies**, discussing their **techniques, tools, challenges, and applications** across domains.

* **Static scraping** - parsing HTML structure using libraries like BeautifulSoup.
* **Dynamic scraping** - handling JavaScript-heavy sites via Selenium.
* **API-based scraping** - leveraging structured endpoints where available.

**2.2 “Web Scraping Techniques to Collect Weather Data in South Sumatera”**

The paper presents a structured approach to web scraping for weather data collection, focusing on South Sumatera, Indonesia. It outlines how publicly accessible weather information from online platforms can be extracted using automated scripts and then stored for analysis. The methodology involves:

* Demonstrates practical, reusable scraping architecture applicable to different regions including India.
* Identifying reliable weather data sources (government and third-party sites).
* Using Python-based libraries (BeautifulSoup, Requests, Selenium) to extract temperature, humidity, wind speed, and rainfall data.
* Cleaning and structuring the data into formats like CSV or database tables for further analysis.

**2.3 “A Comparative Analysis of Data Mining Methods for Weather Prediction"**

This paper evaluates **data mining techniques** for predicting weather parameters such as temperature, rainfall, humidity, and wind speed.

The analysis covers:

* **Data preprocessing** : handling missing values, normalization, and feature selection.
* **Training and testing** : models with past meteorological data.
* **Performance evaluation :**  using metrics such as accuracy, precision, recall, and RMSE.

**2.4 “A Web Scraping Framework for Meteorological Big Data”:**

This paper presents a structured framework for **collecting, processing, and storing large-scale meteorological data** from online sources using web scraping.

* Objective: Automate acquisition of weather data from diverse sources to support forecasting, climate analysis, and research.
* Framework Architecture:
  1. Data Acquisition Layer – Uses scraping tools (e.g., Scrapy, Selenium) to extract data from official meteorological portals.
  2. Pre-processing Layer – Cleans, formats, and validates the raw data, handling missing values and inconsistencies.
  3. Storage Layer – Stores processed data in big data systems like Hadoop, Spark, or cloud databases.
  4. Application Layer – Provides interfaces for data visualization, analysis, and integration with machine learning models.

**2.5 “Weather Report on Metropolitan Indian Cities via Web Scraping”**

This study focuses on gathering up-to-date meteorological data across major Indian metropolitan regions such as the National Capital Region (NCR), Mumbai, Kolkata, Bengaluru, Hyderabad, Chennai, Pune, Kanpur, Visakhapatnam, and Nagpur using automated web scraping techniques.

* Uses Python **- BeautifulSoup, Requests** for weather data scraping.
* Targets **temperature, humidity, wind, feels like, timestamps**.
* Data cleaned & stored in **CSV/data warehouse**.

**2.6 “Data Analysis by Web Scraping using Python”**

This paper introduces a simple yet structured workflow for extracting, processing, and storing online data using Python-based web scraping techniques for analytical purposes.

* **Objective:** Automate the collection of relevant online data and store it for further analysis in research or business applications.
* **Framework Architecture:**

**Data Extraction Layer –** Uses Python libraries **(BeautifulSoup, Requests, Selenium)** to parse and retrieve required information.

**Storage Layer –** Saves the extracted data in **CSV** or database formats for subsequent processing.

**Chapter 3: Methodology**

The methodology employed in this project follows a systematic and modular pipeline designed to enable efficient extraction, cleaning, analysis, and visualization of data from online weather platforms. The goal is to ensure the collected information is reliable, accurate, and ready for downstream applications.

**3.1 Study Area**

The study area for this project is **Haldwani**, a city located in the Nainital district of Uttarakhand, India. Haldwani is situated at the foothills of the Himalayas, characterized by a subtropical climate with distinct seasonal variations including hot summers, a monsoon season, and cool winters.

Due to its geographic location and climatic conditions, Haldwani presents a representative environment for examining meteorological parameters such as temperature, humidity, and wind speed. The weather data collected from this area reflects a mix of tropical and temperate climatic influences, making it ideal for assessing the accuracy and consistency of data sourced from different weather platforms.

The focus on Haldwani allows the project to gather localized weather information, which is useful for region-specific analysis and potential forecasting applications. The city’s varied weather patterns throughout the year provide a comprehensive dataset to test and compare the reliability of weather data from Meteostat, Wunderground, Visual Crossing, Visual Crossing.

**3.2 Data Collection and Sources**

Data is collected from online platforms that offer real-time and historic hourly weather data. In this project, **Meteostat, Wunderground, Visual Crossing** meteorological sites are treated as data sources due to their high update frequency and historical archives. Web scraping techniques are developed using Python to navigate these sites and programmatically extract desired parameters without manual intervention. The scraper modules are designed to capture information such as temperature, wind speed, humidity,timestamps.

**Wunderground** aggregates data from personal weather stations and official meteorological sources (nearset airpost station - pantnagar). Data is extracted by scraping hourly weather tables using Selenium.

**Meteostat** gathers data from official weather stations. Data is accessed by specifying latitude and longitude points.

**Visual Crossing** combines data from official meteorological agencies, airport weather stations, private weather station networks, and global weather models. For Haldwani, it typically uses the nearest airport station.

**3.3 Input Features**

The input features for this project include key meteorological parameters and temporal information collected from Wunderground, Meteostat, Visual Crossing:

* **Date & Timestamp:** The specific date and time of the weather measurement, essential for time series analysis and forecasting.
* **Temperature (°C):** Ambient air temperature at the given time.
* **Humidity (%):** Relative humidity indicating moisture content in the air.
* **Wind Speed (km/h):** Wind velocity at the measurement location.

These features form the basis for data visualization, comparison, and predictive modeling.

**3.4 Data Preprocessing**

Before analysis, the collected weather data undergoes several preprocessing steps to ensure quality and consistency:

* **Data Cleaning:** Missing, duplicate, or inconsistent entries in the datasets from Wunderground and Meteostat are identified and handled.
* **Data Formatting:** Date and timestamp fields are standardized to a uniform format to facilitate time series alignment and merging across sources.
* **Unit Consistency:** Weather parameters are converted to consistent units if needed (e.g., wind speed standardized to km/h).
* **Outlier Detection:** Anomalous values that deviate significantly from typical patterns are flagged and reviewed to avoid misleading analysis.

This preprocessing pipeline ensures the datasets are clean, coherent, and ready for effective visualization, comparison.

**3.5 Tools and Libraries Used**

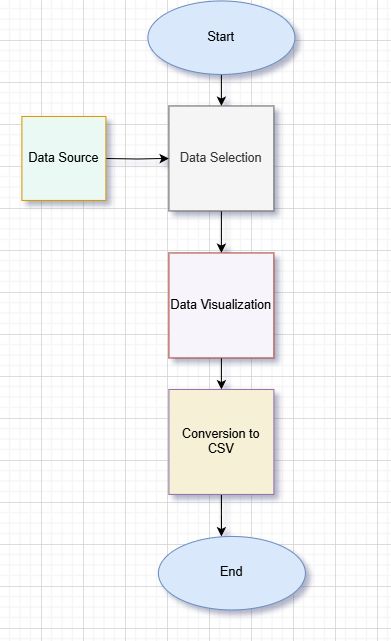
The project uses Python as the main programming language, libraries like Selenium for web scraping, and Requests for HTTP interactions. Data processing is handled with Pandas and NumPy, while Matplotlib and Seaborn are used for visualization. Development takes place in Google Colab, with ChromeDriver.

The following are the combination of programming tools and libraries to extract, process, analyze, and visualize weather data efficiently:

* **Programming Language:** Python
* **Web Scraping Libraries:**  
  BeautifulSoup - for parsing HTML and extracting data from static web pages  
  Selenium - for interacting with dynamic websites  
  Requests - for sending HTTP requests and fetching responses
* **Data Processing & Storage:**  
  Pandas - for data manipulation and CSV export
* **Data Visualization:**  
  Matplotlib, Seaborn - for generating graphs and charts
* **Development Environment:**  
  Google Colab

**3.6 Model Architecture:**

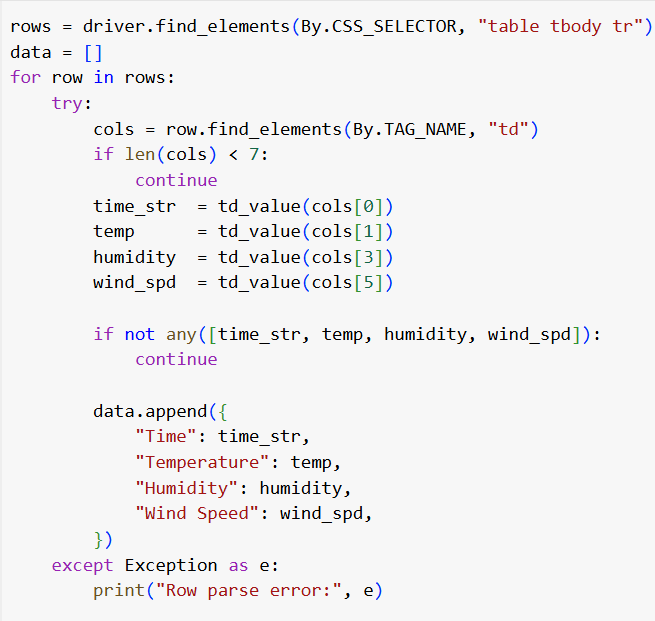
* **Start**
  + System begins execution.
  + Loads configuration parameters such as target city,date.
* **Source Identification & Webpage Loading**
  + Identifies valid URLs from which data will be collected.
  + Requests are sent or Selenium browser is launched to fetch the HTML content of each source.
* **Data Extraction**
  + Specific HTML elements are parsed and target features (e.g., time, temperature, wind speed) are extracted.
* **Data Selection**
  + The raw scraped data is filtered to retain only meaningful attributes.
  + Duplicate or irrelevant entries are removed.
* **Data Visualization**
  + Cleaned data is plotted using time-series line graphs or comparison charts.
  + Helps in trend detection and understanding hourly patterns.
* **Data Conversion**
  + Processed data is exported into structured files.
  + Separate CSVs are created per site.



*Figure 1. WorkFlow.*

**3.7 Program Implementation**

This code segment is responsible for programmatically retrieving hourly weather data from the Wundergro.und website. It uses the Selenium automation framework to simulate a browser session, navigate to the appropriate webpage, and identify table elements that contain weather metrics such as temperature, humidity, wind speed, and time. The scraper carefully locates table rows using CSS selectors, then extracts relevant columns from each row. It includes checks to skip incomplete rows and handles hidden or attribute-based text using a custom parsing function. The result is a structured collection of clean weather data that can be stored or visualized.



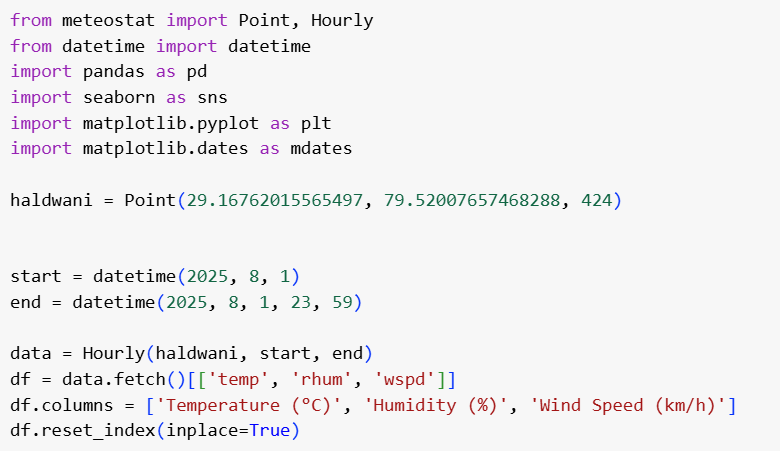
*Figure 2. wunderground scraping code*

**Helper Function:**



*Figure 3. wunderground helper function*

* The following code snippet demonstrates how historical hourly weather data is programmatically extracted from the **Meteostat**. Using precise geographical coordinates (latitude, longitude, and elevation), this script fetches **temperature**, **humidity**, and **wind speed** data for a specified time range. The retrieved data is then cleaned, formatted into a structured DataFrame, and saved as a CSV file for further analysis and visualization.



*Figure 4. Meteostat scraping code*

* This code retrieves hourly weather data from Visual Crossing by sending an HTTP request to the specified URL. The JSON response is parsed to extract key parameters such as date, time, temperature, humidity, wind speed, and conditions. These values are stored in a structured list, converted into a Pandas DataFrame, and saved as a CSV file named according to the target date and location (Haldwani). This approach ensures the weather data is organized in a clean, consistent format for further analysis and visualization.



*Figure 5. Visual Crossing scraping code*

**Chapter 4: Result and Discussion**

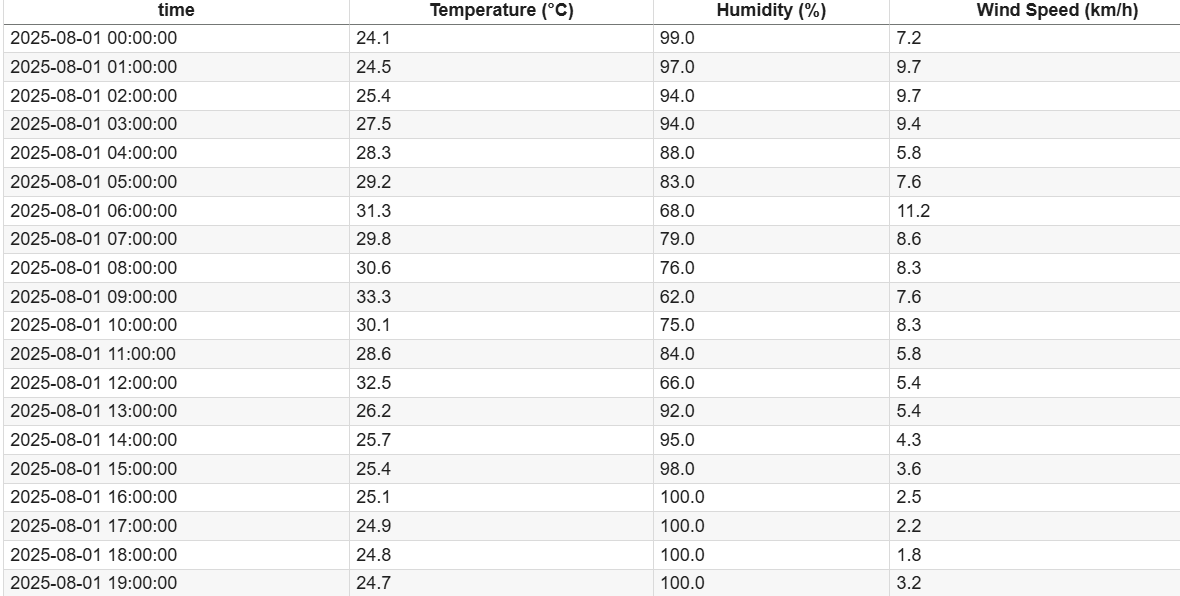
The project successfully extracted weather data for Haldwani from Wunderground, Meteostat, Visual Crossing for the specified time period. The data collected included temperature, humidity, and wind speed, which were cleaned and organized into CSV files for visualization.

**4.1 Result and Outcome**

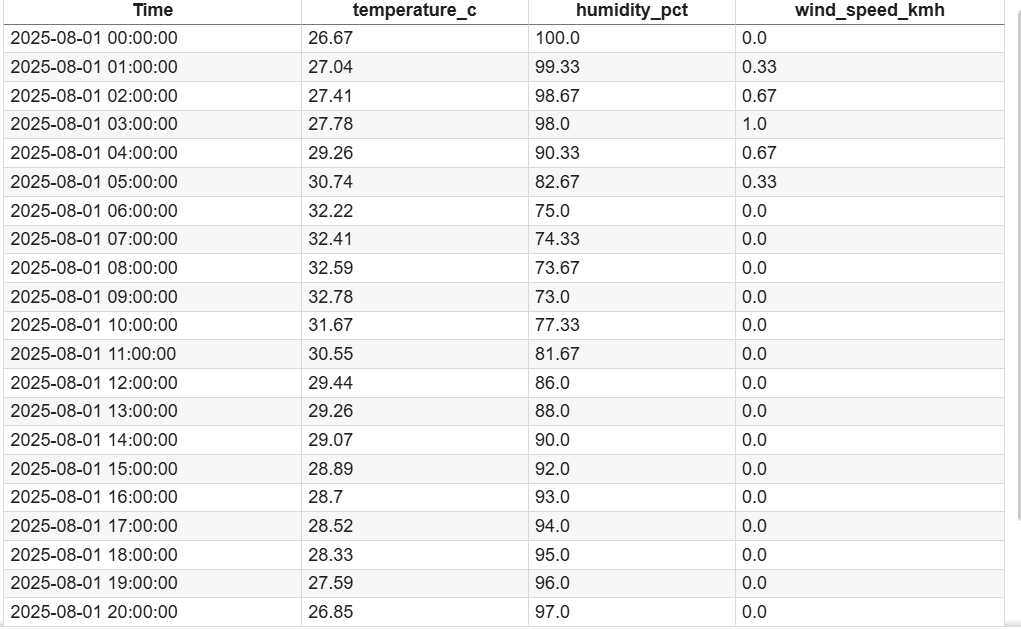
The **Smart Scraper** system is successfully designed and deployed to retrieve and process real-time hourly weather information for the defined study area. The scraper modules interfaced with online sources including Meteostat, Wunderground, Visual Crossing and were able to autonomously extract atmospheric parameters such as time, temperature, humidity, wind speed.

Once the raw tables were extracted, the data passed through a preprocessing pipeline where formatting conflicts, unit mismatches, and missing values were resolved. The processed datasets were saved in a clean, machine-readable format for analysis. Representative snippets from this process are summarized below:

* **Meteostat Processed Dataset**  
  This table showcases the hourly weather measurements obtained from the Meteostat platform. The dataset comprises timestamped observations of temperature (°C), relative humidity (%), and wind speed (km/h) across a 24-hour period. The clean and structured output demonstrates how the Smart Scraper can seamlessly integrate data from sources along with scraped sources



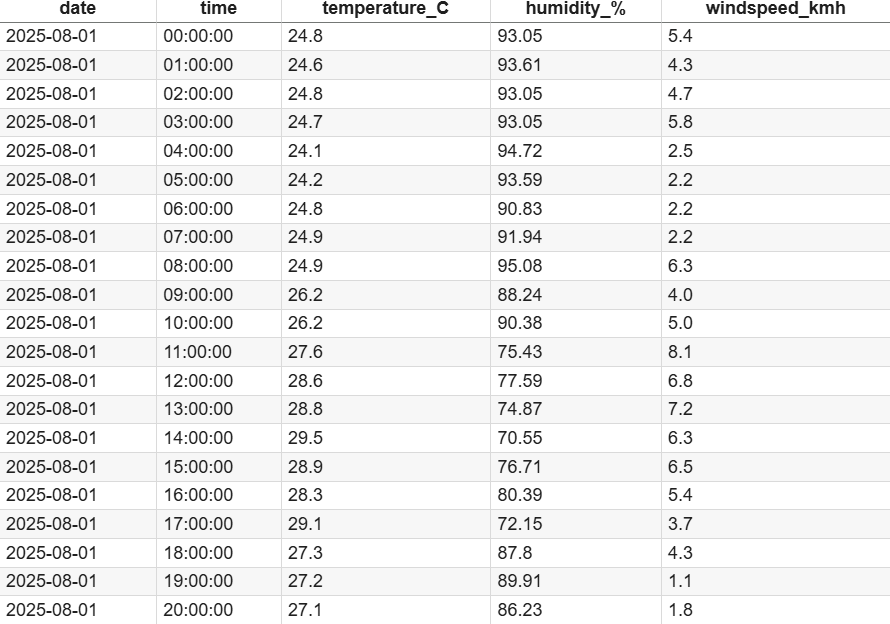
*Figure 6.Meteostat Processed Dataset*

**Wunderground Processed Dataset**  
The data is provided in a numeric, analysis-ready format including temperature in °C, relative humidity in percent, and wind speed in km/h, minimizing the need for additional preprocessing. Such clean and consistent data from Wunderground is useful not only for visualization, but also as a reliable ground-truth reference to validate information scraped from other websites. 

*Figure 7. Wunderground Processed Dataset*

* **Visual Crossing Dataset**

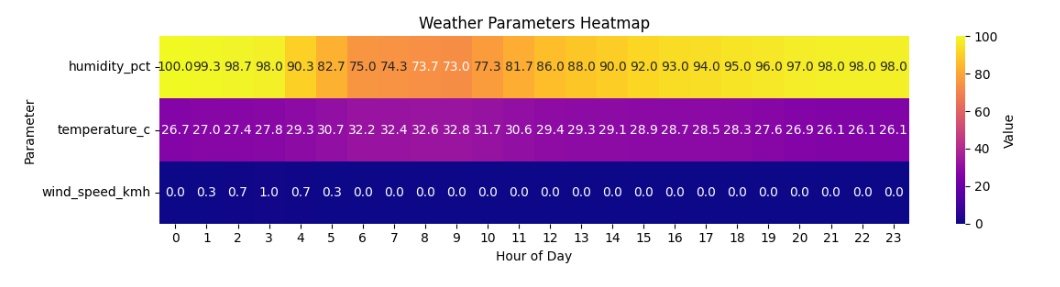
The dataset shows hourly weather for **1 August 2025**, with temperature rising from about **24.6°C** at midnight to **29.5°C** in the afternoon, then cooling by evening. Humidity is highest in the early morning, dips at midday, and rises again later, while wind speed peaks around late morning and is lowest in the early morning and late evening.



*Figure 8. Visual Crossing Processed Dataset*

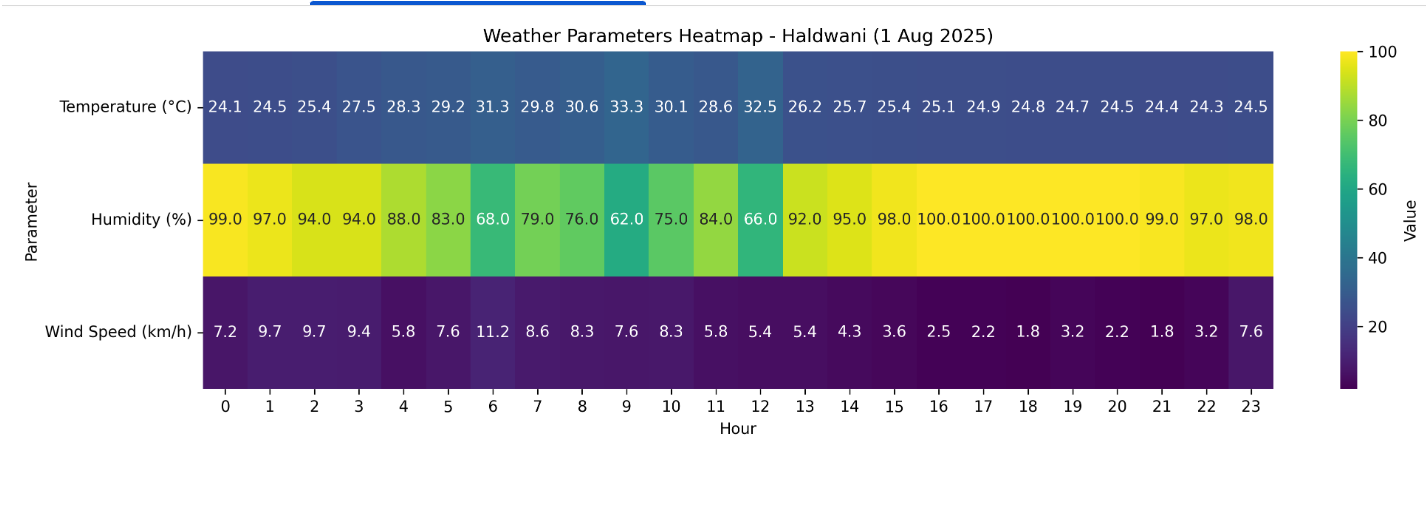
**4.2 Visualization**

This visualization represents hourly weather data from **Wunderground** for **Haldwani on August 1, 2025**. It includes key parameters such as **temperature, humidity**, and **wind speed**. **Heatmaps** offer a clear visual of hourly trends and provides detailed insights into Haldwani’s local weather conditions.



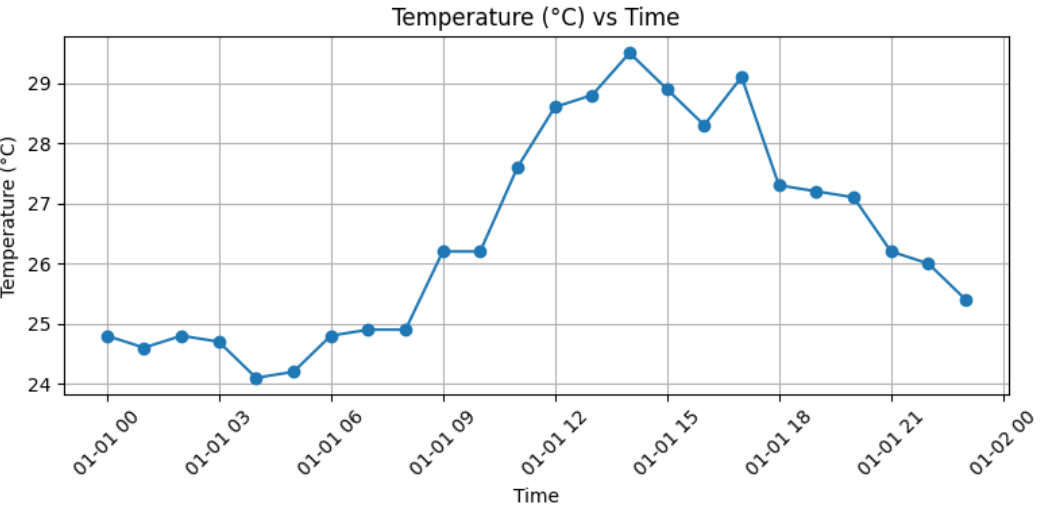
*Figure 9. Wunderground plot*

This visualization displays **hourly weather data from Meteostat** for **Haldwani on August 1, 2025**. The dataset includes parameters such as **temperature, humidity**, and **wind speed**, extracted using the city's **latitude and longitude coordinates**. **Heatmaps** highlight hourly patterns in a more intuitive way.

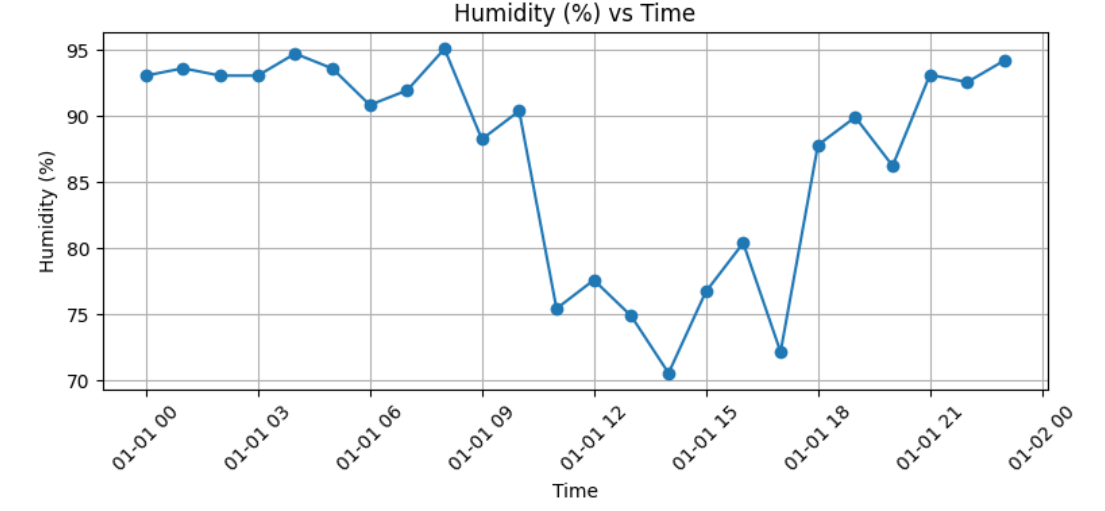


*Figure 10.Meteostat plot*

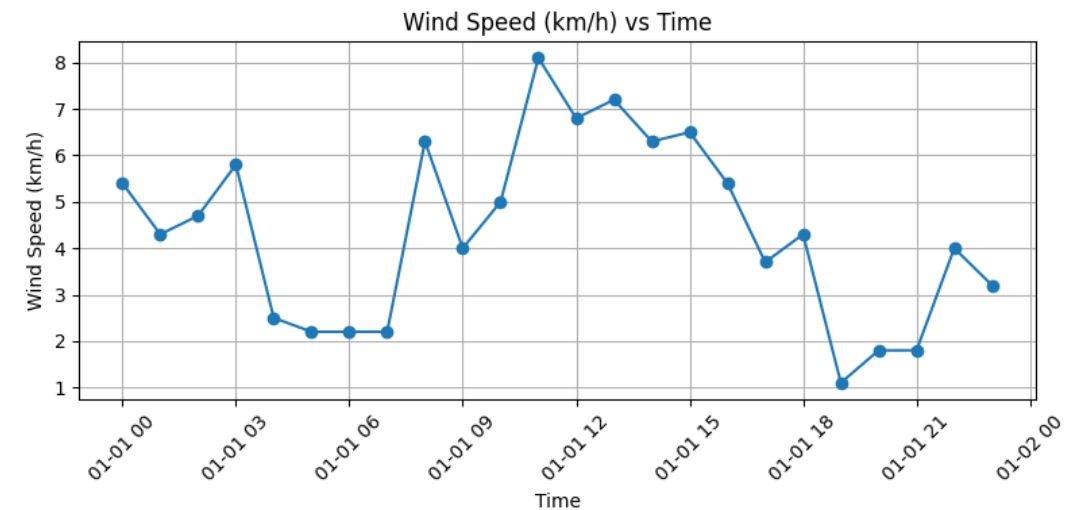
This visualization represents hourly weather data from **Visual Crossing** for **Haldwani on August 1, 2025**. It includes key parameters such as **temperature, humidity**, and **wind speed**. **Line plot** offer a clear visual of hourly trends and provides detailed insights into Haldwani’s local weather conditions.



*Figure 11. Visual Crossing Temperature Plot*



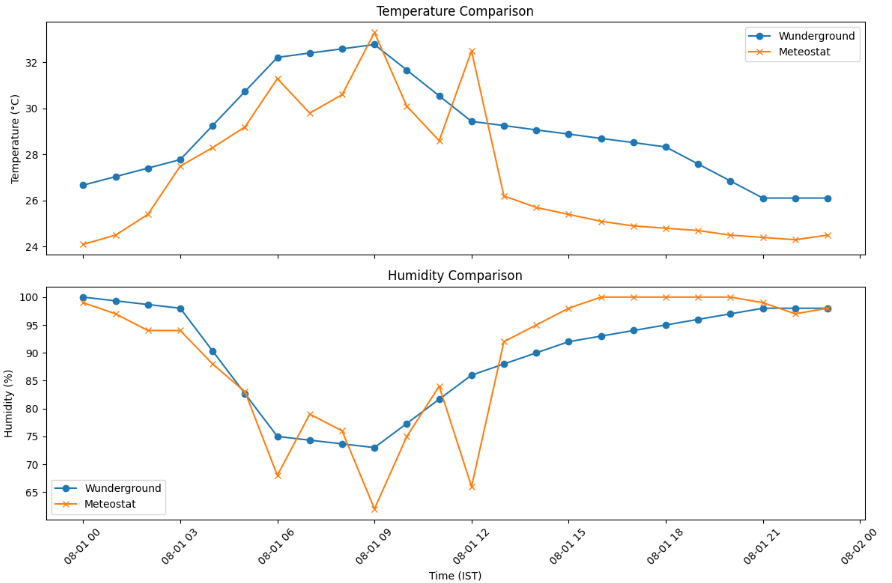
*Figure 11. Visual Crossing Humidity Plot*



*Figure 11. Visual Crossing Wind Speed Plot*

**4.3 Comparative Analysis**

This presents a comparative analysis of weather data obtained from two independent sources: **Wunderground** and **Meteostat, Visual Crossing**, specifically for the city of **Haldwani** on **August 1, 2025**. The data from both sources were aligned based on timestamps and merged to enable direct hour-by-hour comparison of key weather parameters. For this analysis, **temperature** and **humidity** were considered, as they are critical indicators of local climatic conditions.



*Figure 9. Comparison plot*

Slight fluctuations in the recorded values between Wunderground, Meteostat, Visual Crossing are observed, which are expected due to differences in their data sourcing methods.

**Meteostat** retrieves weather data based on **geographical coordinates (latitude and longitude)** and sources it from standardized meteorological stations, often at regional airports. In contrast, **Wunderground** relies on data from **Personal Weather Stations (PWS)** and nearby official stations such as airports. This can lead to **inconsistencies** in temperature, humidity, or wind speed readings due to differences in **sensor location, altitude, and microclimate conditions**. **Visual Crossing** sources its weather data from a combination of official meteorological stations, satellite observations.

**Chapter 5: Summary and Conclusions**

### **5.1 Summary**

This project aimed to collect, process, and analyze weather data from different online sources **Wunderground**, **Meteostat, Visual Crossing** -for the city of **Haldwani**. Key meteorological parameters such as **temperature**, **humidity**, and **wind speed** were extracted using a combination of web scraping.

Selenium and Requests were used to scrape content from Wunderground, while the Meteostat Python library provided structured historical weather. All extracted data was cleaned, standardized, and stored in **CSV format**. Visualizations were created to identify trends and assess consistency between the two datasets.

The results showed strong similarity in temperature and humidity trends, with minor variations in wind speed values. These differences were attributed to the use of different data sources Wunderground uses personal weather stations, while Meteostat relies on official weather stations like **Pantnagar Airport**.

### **5.2 Conclusions**

### This project successfully demonstrates the automated collection and analysis of weather data from distinct sources: Wunderground, Meteostat, Visual Crossing. By leveraging web scraping techniques, key parameters such as temperature, humidity, and wind speed were extracted and consolidated into clean, structured datasets. The comparative analysis showed strong consistency in temperature and humidity values, validating the reliability of both sources. Minor differences in wind speed, temeprature and humidity highlights the importance of using data sources for cross-verification.

The preprocessing and formatting of data into CSV files enabled effective visualization and laid the foundation for future forecasting models. Incorporating date and time stamps allowed for temporal analysis essential in weather prediction. Overall, the project presents a scalable framework that can be adapted to other locations and expanded to include additional weather parameters. This approach enhances the accuracy and reliability of weather data analytics by integrating diverse datasets and automated extraction methods.

**Chapter 6: References**

[1] R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso and S. N. Mbaye, “Web Scraping: State-of-the-Art and Areas of Application,”*2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 6040-6042, doi: 10.1109/BigData47090.2019.9005594.

[2] Fatmasari, Y. N. Kunang and S. D. Purnamasari, “Web Scraping Techniques to Collect Weather Data in South Sumatera,”*2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Pangkal, Indonesia, 2018, pp. 385-390, doi: 10.1109/ICECOS.2018.8605202.

[3] M. Zeyad and M. S. Hossain, “A Comparative Analysis of Data Mining Methods for Weather Prediction,”*2021 International Conference on Computational Performance Evaluation(ComPE)*,Shillong,India,2021,pp.167172, doi:10.1109/ComPE53109.2021.9752344.

[4] El Mhouti, Abderrahim & Fahim, Mohamed & Soufi, Adil & Alama, Imane. (2022). A “Web Scraping Framework for Descriptive Analysis of Meteorological Big Data for Decision-Making Purposes,” *International Journal of Hybrid Information Technology.* 2. 47-64. 10.21742/ijhit.2653-309X.2022.2.1.04.

[5] Journal, Ijariit. “Weather Report on Metropolitan Cities in India Using Web Scraping Technique.” *International Journal of Advance Research, Ideas and Innovations in Technology*, 2019.

[6] D. M. Thomas and S. Mathur, “Data Analysis by Web Scraping using Python,”*2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.