BUILDING A WEATHER FORECASTING APPLICATION USING API

Premananda Sahu, Ayush Mishra
Lovely Professional University, Punjab, India
Corresponding Author
Vishal Rana
Lovely Professional University, Punjab, India

ABSTRACT- It has always been a difficult task to find a location specific high quality past and near future data for a building community while performing the research or predicting the weather control. In this study, we provide a unique approach: the usage of a weather generator as a weather API. The benefits of this approach are illustrated with a few examples of applications. In conclusion, a weather API makes it easier to integrate models into frameworks, reduces the amount of maintenance needed, and avoids issues with interoperability caused by different computer languages. The data in the Capstone Weather App is shown using the OpenWeatherMap API. Academics and programmers of web-based services and mobile applications may access weather data, including predictions and current analysis data, using the OpenWeatherMap API, an online meteorological service. It might take a lot of effort and money to create and update REST API documentation with use examples for APIs that are continually evolving. Despite the fact that usage examples must be manually entered, several REST API documentation solutions focus on automating API object documentation. By following the suggestions from this paper's REST API documentation, REST API developers may reduce mistakes, boost success rates, and make API client developers happier.

Keywords: Weather data, weather API, Capstone Weather App, OpenWeatherMap API, REST API, meteorological.

INTRODUCTION

Because it enables us to organise our activities according to the weather, weather forecasting has become an important element of our everyday life. Technology improvements have made it simpler to acquire real-time weather data and offer reliable weather forecasts. In this project, we used API to create a weather forecasting application. Real-time meteorological information for a given place is obtained by the programme via a REST API, and it is then presented in an intuitive manner. The user may enter their location to obtain weather information specific to that area. The Web's architecture is described by the software architectural style known as "Representational State Transfer" (REST). The following steps are included in the project: To get an API key, we first decided on a weather API provider and registered for the API. To obtain current weather information for a specified area, we performed API calls to the provider's REST API. To extract the necessary weather, we parsed the JSON data that we got from the API. The programme offers extra features including weather warnings, hourly or daily predictions, and weather history in addition to presenting real-time weather data. In order to be informed when

the weather changes, the user can set up weather notifications. The hourly or daily forecasts give a prediction of the weather for the upcoming few hours or days for a certain place. The weather history gives a chronological account of the weather at a certain area over an extended period of time.

To study the weather, scientists utilise devices referred to as weather stations. These weather stations are scattered over the globe, collecting data on various occurrences. Several weather groups have begun putting thousands of these sensors into massive networks to collect meteorological data. These gadgets capture meteorological data and transmit it to the collecting points for subsequent processing. Every method used to transmit meteorological data, however, is dependent on protocols that were not intended for this use. Due to the data formats and protocols used in them, the weather stations are unable to utilise the real-time data that is available on them.

In conclusion, the user-friendly weather forecasting application created utilising API offers precise weather data and forecasts for a particular place. Users can use the programme using a web or mobile interface. Individuals, companies, or organisations can use the programme to arrange their activities based on the weather. The project may be expanded to incorporate other elements, such as satellite images and weather maps. The project shows the strength of APIs and how they can deliver real-time data for a variety of applications.

RELATED WORK

The ability to receive real-time data from numerous sources in real-time is one of the key benefits of employing API technology in weather forecasting. By combining information from several sources, this enables weather forecasts to be more precise. Additionally, weather forecasting programmes may be readily connected with other software systems, such as transportation systems and emergency response systems which is only possible because of weather API.

A number of research investigations and development initiatives have been undertaken with the goal of creating weather forecasting apps that leverage API technology. One such initiative is the OpenWeatherMap API, which gives programmers access to current weather information for a particular place. Numerous weather forecasting applications, especially those for mobile devices and the web, frequently use the OpenWeatherMap API.

Another study looked at how to increase the accuracy of weather forecasts using machine learning algorithms and API technologies. A machine learning model was created as part of the project to analyse weather information from API sources and produce more precise weather forecasts for a given region. The study demonstrated that utilising machine learning algorithms with API technologies might greatly increase the accuracy of weather forecasts.

Many researchers and publishers concluded about the weather forecasting using API as follows:

O. M. Brastein et.al [1], [2], [3] in the architecture based on weather API told about weather prediction utilising an api to implement a classification model in a weather forecasting application. The filtering of datasets from open weather to weather forecasting api is one of the

strategies used in this article. The API package of Python provides an ease to get the free weather data. A bunch of weather data providers is supported by the package and other providers can be added in future.

E. B. Abrahamsen et.al [4], [5] on Weather Prediction stated that Artificial neural networks are utilised in this study to forecast temperature. In order to forecast the temperature 1, 3, 6, and 12 hours in advance, four different models were trained. In the first Experiment, temperature was the only input provided to the networks. An auto-regressive neural network (AR-NN) is what this is. In the second experiment, a network formed by an autoregressive neural network with exogenous was given precipitation data.

Rung-Ching Chen et.al [6], [7] stated that the OpenWeatherMap website provided all of the real-time weather information used in this article. Utilising basic Python programming ideas, this weather data will be sent to Twitter. The Python code can interface with Twitter and use its API due to OAuth and Tweepy, a useful module. In this experiment, we may obtain real-time weather information such as the temperature (maximum and lowest), wind speed, humidity, cloud cover, pressure, dawn and sunset times, and current weather in a particular city and nation. Anyone looking to use the weather may benefit from tweets about it.

B Bochenek and Zbigniew Ustrnul [8] stated about Apps for weather forecasting have been the subject of several research studies and development initiatives. These initiatives employed API technology to create these apps. One such initiative is the OpenWeatherMap API, which gives programmers access to current weather information for a particular place. Numerous weather forecasting software programmes, including web-based and mobile weather programmes, make extensive use of the OpenWeatherMap API that API sources supplied more precise weather predictions for a particular place.

METHODOLOGY

The following steps are used to achieve the objectives of the Project:

1. Building a GUI Application

We have used python and its various libraries to build a Weather Forecasting GUI Application to show the weather data of a specific city in a User-friendly way. The flowchart for designing the GUI for this research is shown in Figure 1. To start, enter "guide" in the Tkinter command window to bring up the GUIDE Quick Start window. The user has four alternatives while building a new GUI. The user is given a blank GUI and a list of available items on the left side of the window. The finished design GUI is then represented utilising the elements that were given. After that, Tkinter's editor window automatically creates the coding for the stored GUI.

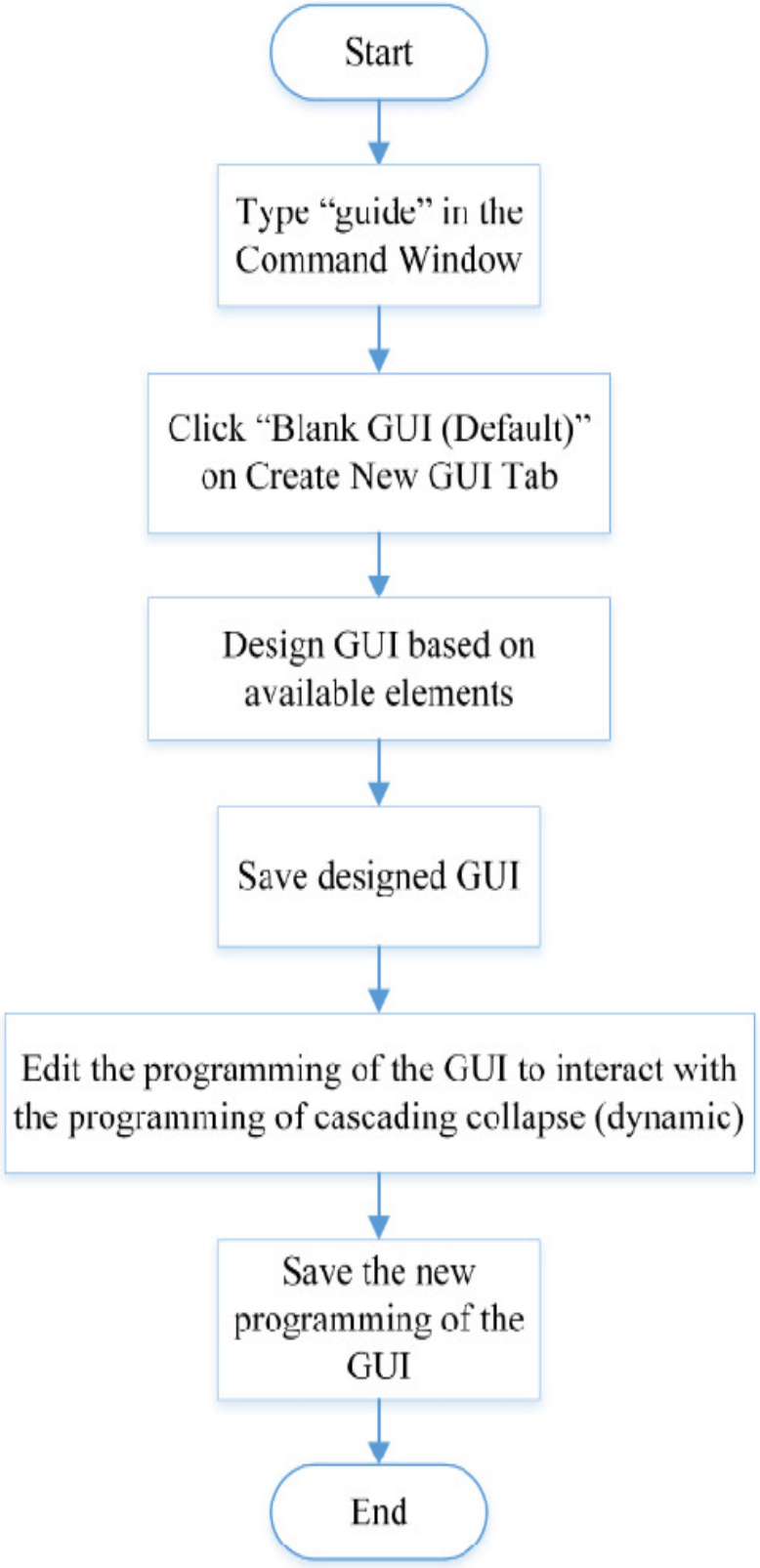


Figure No. 1: Building of Application.

2. Making a API Connection

API connection is the essential and core component of our GUI Application. We have made the connection of the GUI with OpenWeatherAPI which provides us the Real Time weather data of a specific city. The Search Bar of the application accepts a city name of any part of the world and provides or displays the accurate weather details of that specific location.

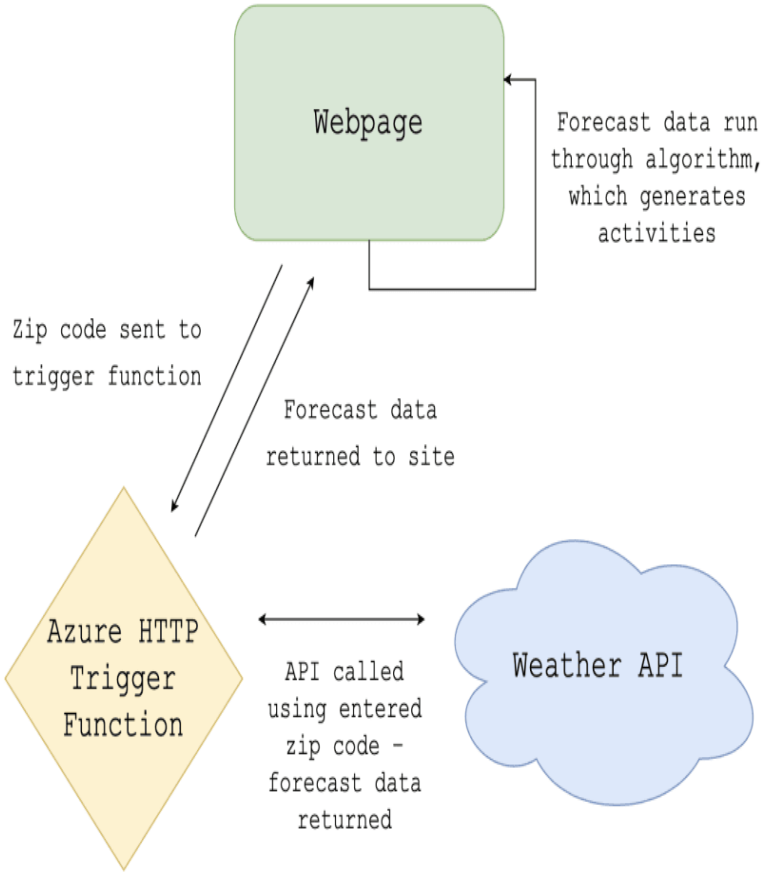


Figure No. 2: Connecting to the API.

3. Retrieving the Real Time Weather Data

As we enter the city name in the search bar option of GUI application, our API retrieves or fetches the corresponding weather data of that specific location which is available and Displays it on the GUI.

4. Displaying the Data.

Tkinter, pandas, json and many other python libraries has helped us in presenting the retrieved weather data in meaningful way with images and icons. We make certain comparisons with the fetched Weather data the actual Weather data to check the error between the two.

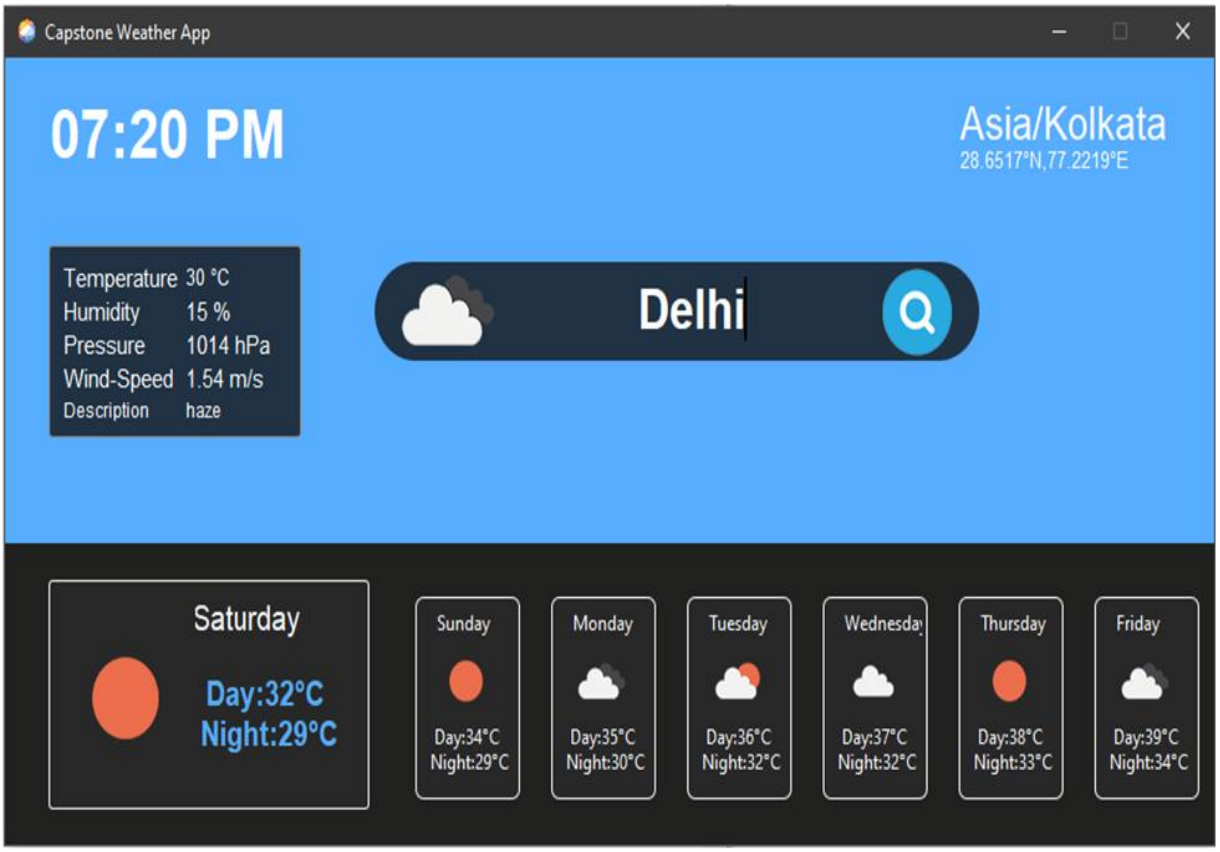


Figure No.3: Getting data.

RESULT & DISCUSSION

1.1 Building a User Interface for Displaying Weather Data.

1.1.1 Choosing a user interface framework (e.g., Tkinter, PyQt, Flask)

Python is a popular programming language: Python is widely used for developing various applications, including web development, machine learning, scientific computing, and data analysis. Its popularity makes it easier to find resources, libraries, and tools for development, as well as to hire developers if needed.

Rich set of libraries and frameworks: Python has a vast collection of libraries and frameworks that can be used for building different applications. For instance, there are many libraries for processing and analyzing weather data, such as NumPy, Pandas, Matplotlib, and SciPy.

Tkinter is easy to use: Tkinter is the standard Python library for creating GUI applications. It is easy to learn and use, making it a great choice for beginners who want to create simple GUI applications.

Cross-platform compatibility: Python and Tkinter applications can be run on different platforms, including Windows, Mac, and Linux, without the need for extensive modifications.

Weather forecasting is a popular application: Weather forecasting is an application that many people find useful, and a well-designed weather forecasting application can be a valuable tool for many users.

4.1.2 Designing the interface layout and components.

Designing a user interface for a weather forecasting application involves several considerations, such as the type of information to be displayed, the layout of the interface, and the visual style of the application. Here's an example of how you might design the interface for a weather forecasting application using Python tkinter:

Choose a layout: There are several options for the layout of the application, such as a single screen or multiple tabs. A good layout should be easy to navigate and should allow users to access the information they need quickly. In this example, we will use a single-screen layout.

Choose the components: The components of the application should be selected based on the information that needs to be displayed. For a weather forecasting application, the following components may be included:

Location selection: Allows users to select the location for which they want to view the weather information.

Current weather information: Displays the current weather conditions for the selected location, such as temperature, humidity, wind speed, and precipitation.

Weather forecast: Displays the forecasted weather conditions for the selected location over the next few days.

Additional information: Allows users to access additional weather-related information, such as UV index, air quality, or radar maps.

Visual style: The visual style of the application should be consistent and easy to read. A good visual style will make the application more appealing to users and enhance the user experience. In this example, we will use a clean and modern design with a blue and white color scheme [9].

4.1.3 Dynamically updating weather data in the interface.

To dynamically update weather data in the interface of a weather forecasting application using Python, we can use the after() method in tkinter. The after() method schedules a function to be called after a specified delay, and can be used to periodically update the weather data in the interface.

When the tkinter interface is run using root.mainloop(), the update_weather() function is called to initially populate the labels with weather data. After 5 minutes, the after() method calls the update_weather() function again, which updates the labels with new weather data, and so on.

By using the after() method in this way, the weather data in the tkinter interface is dynamically updated without the need for the user to manually refresh the page or interact with the application [10].

4.1.4 Displaying weather data using images, icons, and graphs.

To display weather data using images, icons, and graphs in a weather forecasting application using Python tkinter, we can use various libraries and tools. Here are a few examples:

Displaying weather icons: we can use the Weather Icons library to display weather icons based on the current weather conditions. The library contains a set of icons for various weather conditions such as cloudy, sunny, rainy, etc. we can use the library to display the appropriate icon based on the current weather conditions in your application.

Displaying weather data using images: we can use various Python libraries such as Pillow, Matplotlib, or Seaborn to display weather data using images and graphs. For example, one can use Pillow to display a satellite image of the current weather conditions, or Matplotlib to display a graph of temperature trends over time.

1.2 Discussing output: Approaching the output, Capstone Weather App displays the time according to the time zone, latitude and longitude, humidity, wind speed, pressure and indeed 7 days temperature as shown in figure.

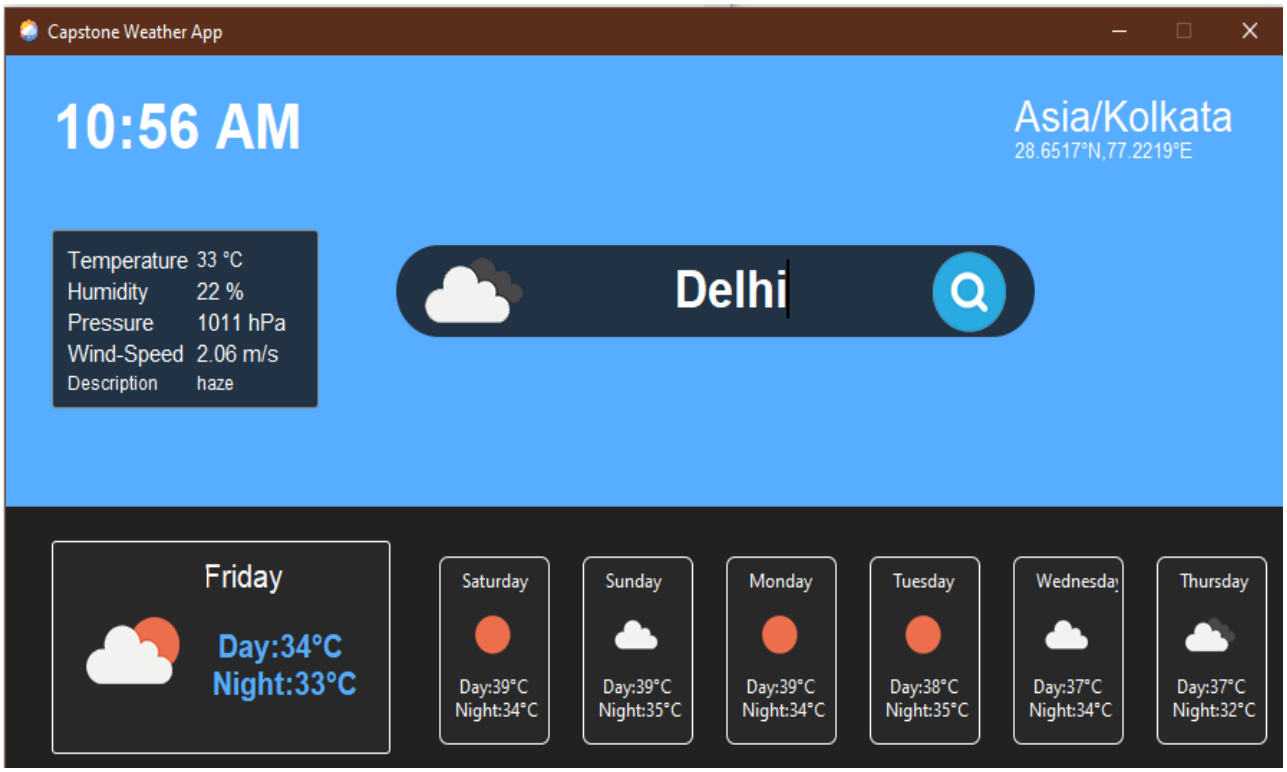


Figure No. 4: The resultant data.

Selecting the weather forecasting menu on the home page is the first step in forecasting the weather data. The system then provides the data for the next 7 days' prediction on the same page. Results of the data are shown in the following table.

Day	City	Max Temp	Min Temp	Humidity	Pressure	Wind Speed
1	Delhi	34	23	22%	1012	2.05
1	Kolkata	35	26	18%	1005	1.02
2	Delhi	32	21	25%	1015	2.15
2	Kolkata	34	24	21%	1009	1.29
3	Delhi	34	22	16%	1011	1.32
3	Kolkata	33	24	18%	1006	1.11

Table No. 1: Output of first three days.

1.3 Error Calculation: The system retrieved the temperature from OpenWeatherAPI which was compared with the original data. The error in the result was shown in the following graph.

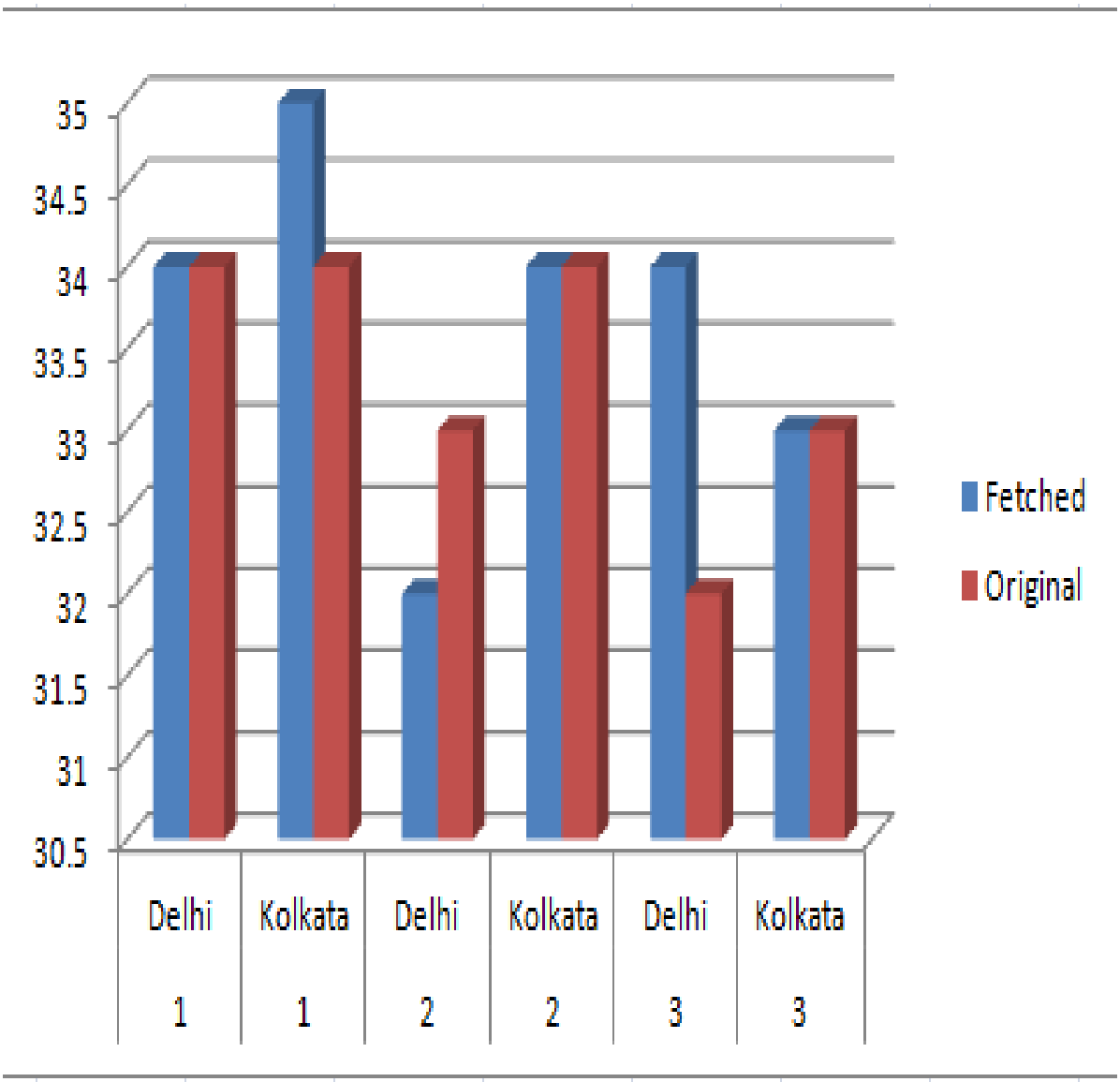
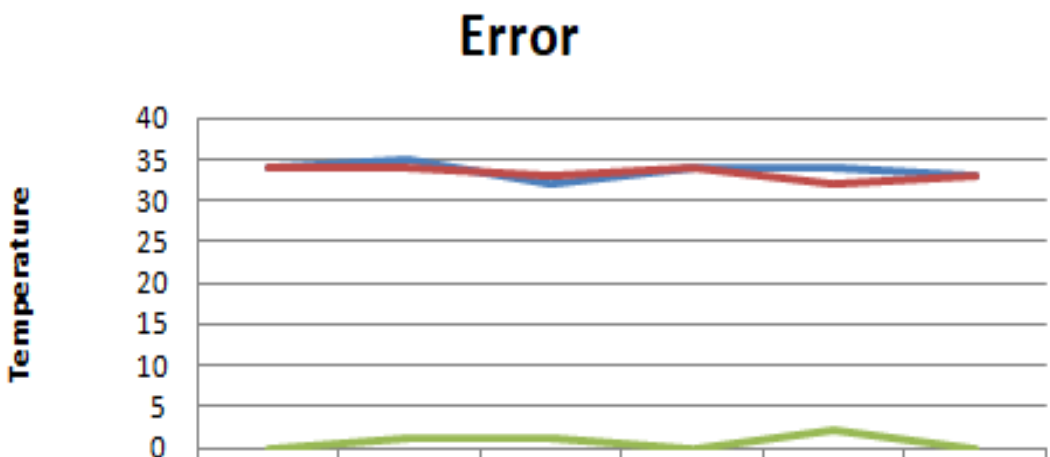


Figure No. 5: Comparison of the data.



	Delhi 1	Kolkata 1	Delhi 2	Kolkata 2	Delhi 3	Kolkata 3
Fetched	34	35	32	34	34	33
Original	34	34	33	34	32	33
Error	0	1	1	0	2	0

Figure No. 6: Error in the data.

1.4 Accuracy of weather data: This metric would indicate how accurate the weather data is that is being retrieved from the API. It could be measured using a comparison between the forecasted weather data and the actual weather conditions over a period.

1. **User satisfaction:** This metric would indicate how satisfied users are with the application. It could be measured using surveys or user feedback forms to gather user opinions on the application's interface, ease of use, and overall functionality.
2. **Performance:** This metric would indicate how well the application performs in terms of speed and reliability. It could be measured using load testing and stress testing to identify any performance bottlenecks or issues.
3. **Adoption rate:** This metric would indicate how widely the application is being used by users. It could be measured using download or installation statistics, or by analyzing user engagement metrics such as active users or daily usage.
4. **Error rate:** This metric would indicate how frequently errors occur in the application, such as failed API calls or exceptions. It could be measured using error tracking tools or by analyzing log files to identify common errors and their frequency.

CONCLUSION and FUTURE WORK

In conclusion, building a weather forecasting application using REST API in Python is a valuable and practical project that can provide users with reliable weather information.

Overall, building a weather forecasting application using REST API in Python is an exciting and rewarding work that can provide valuable information to users. By following best practices and

taking key considerations into account, developers can create a reliable, accurate, and user-friendly application that meets the needs of their target audience.

We've given the first formal description of JSON Schema's syntax and semantics in response to the issues caused by the lack of a formal specification for it. We have carried out a rigorous analysis of the schema validation issue and presented effective solutions. We test our own validation tool to demonstrate the practical application of JSON Schema and demonstrate that it performs well both when processing artificially generated data and when tested against the whole Wiki data database.

If the errors are going to be deducted by choosing the latest API methods and fix the issues for getting the weather report while having no access to the internet, this will be referred as our future work.

REFERENCES

1. Brastein, O.M., Perera, D.W.U., Pfeifer, C. and Skeie, N.O., 2018. Parameter estimation for grey-box models of building thermal behaviour. Energy and Buildings, 169, pp.58-68.
2. Abrahamsen, Erik, Ole Magnus Brastein, and Bernt Lie. "Machine learning in python for weather forecast based on freely available weather data." In Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway, no. 153, pp. 169-176. Linköping University Electronic Press, 2018.
3. S. Zhang, W. Wang, X. Gao, and C. Liu, "Design and implementation of weather forecasting service based on RESTful web service," in 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), Vietri sul Mare, Italy, 2013, pp. 428-434.
4. Selvik, J. T., Bansal, S., & Abrahamsen, E. B. (2021). On the use of criteria based on the SMART acronym to assess quality of performance indicators for safety management in process industries. Journal of Loss Prevention in the Process Industries, 70, 104392.
5. Abrahamsen, E. B., Selvik, J. T., Dahle, A. N., Asche, F., & Abrahamsen, H. B. (2018). A socio-economic analysis of increased staffing in the Norwegian helicopter emergency medical service. Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine, 26, 1-9.
6. Chen, Rung-Ching, et al. "Selecting critical features for data classification based on machine learning methods." Journal of Big Data 7.1 (2020): 52.
7. Liu, Lijuan, Rung-Ching Chen, and Shunzhi Zhu. "Impacts of weather on short-term metro passenger flow forecasting using a deep LSTM neural network." Applied Sciences 10.8 (2020): 2962.
8. Bochenek Bogdan, and Zbigniew Ustrnul. "Machine learning in weather prediction and climate analyses—applications and perspectives." Atmosphere 13.2 (2022): 180.

9. P. Sahu, S. Kumar Mohapatra and P. Kumar Sarangi, "Forecasting of Statistical Crisp Weather in India by Exploration of Different Machine Learning Techniques," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1527-1530, doi: 10.1109/ICACITE53722.2022.9823868.
10. P. Sahu, S. Kumar, R. Ahuja and A. Kaur, "Forecasting of Precipitation in India by Different Data Types using Investigation of Radial Basis Function Neural Network Model," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596073.