**Project Report**

**Weather Data Visualizer**

Group # 3

***Declaration***

We, the **Group # 3**, declare that the project report titled **Weather Data Visualizer** results from our group effort. We affirm that we have not copied or plagiarised the project content from any source. Any external descriptions, concepts, or information used have been appropriately rephrased in our own words, and proper references have been provided where applicable.

We understand and agree that this project report will be subjected to a **Turnitin** check to verify its originality.

|  |  |  |
| --- | --- | --- |
| **Name of the Students** | **Username** | **Student ID** |
| Diego Bernal | d\_bernalolaya |  |
| Xavier Merino | x\_merinomino | 1254901 |

**Abstract** *(100 to 150 words) = (heading font size = 18, font = Times New Roman = it’s applicable for all headings)*

Summary of the project’s purpose, methodology, and findings. *(Font size = 12, font = Times New Roman = it’s applicable for all descriptions)*

**Introduction** *(half page highest, or 300 to 350 words)*

Problem Statement. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

Motivation for the project. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

Overview of the implemented solution. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

**Environment Setup**

A short description *(~50 words)*

*Then………..*

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

Description of tools and technologies used

*(e.g., Python version, libraries, APIs, IDEs, etc.).*

Detailed steps for environment setup, including installation and configuration of dependencies.

* Describe
* Add figures

**Libraries and Utilities**

A short description *(~50 words)*

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

*Then………..*

**Table of Libraries Used**: *(one example given)*

**Table Title**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Purpose** | **Reason for Choice** |
| Matplotlib | Data Visualization | High-quality plotting for graphs and charts |

Justification for selected libraries and their relevance to the project goals.

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

**Implementation**

A short description *(~50 words)*

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

*Then………..*

*If any formula or equation you are going to write in any section of the report, make sure you followed the following rule:*

*a**b* 

Note that the equation is centred using a centre tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

Functions Details: *(that you created and defined for the project purpose)*

**Table Title**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Purpose** | **Reason for Choice** |
| visualize\_data() | Plots data trends | Reads cleaned data and plots graphs |

Code snippets for key functions *(if applicable)*.

Step-by-step implementation process.

**Results and Analysis** *(in total 300 to 350 words)*

A short description *(~50 words)*

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

*Then………..*

Graphs and charts showcasing key findings *(e.g., visualisations generated using Matplotlib, Seaborn, Plotly, etc.)*.

Interpretation of results.

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

Explanation of how results align with the project objectives.

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2].

**Challenges and Solutions**

The main challenges and solutions related to the app development were visual development and API use. The team has more experience in backend development; thus, it was needed to check the documentation and do some tests for the interface. In addition, for the API, there are some paid weather APIs, so it was needed to look for an open and easy-to-use API.

To sum up, the key challenges faced during the implementation of the project were the following:

* The creation of an interactive data app using the Streamlit library.
* Find and use the appropriate API to get open weather data.
* Manipulate the data inside the app.

The solutions adopted to overcome these challenges were the following:

* Learn how to code with the library using the official documentation, [1] a tutorial [2], and a step-by-step tutorial [3].
* Look for an easy, open, and robust API to get the weather information. In this case, Open-Meteo [4].
* Use code and techniques learned in Python and Data Science class.

**Limitations**

The Streamlit library has limited scope for creating personalized UI. It doesn’t offer wide customization or components. For simple projects, the library is robust and easy to use, but for more complex applications, it is limited and not recommended.

The API, in its open use, can only get data for 90 days. To get its full functionality, it has a price of 43.31 CAD monthly. For the current project, it was used with the open data. In addition, the humidity data is stored by hour. To see this data for a longer period, it is necessary to perform an average or another math operation depending on the objective.

**Lessons Learned**

The project pipeline and teamwork were key to the successful completion of the current project. Beyond the review of documentation, coding, testing process, and class presentation, a solid process guided us to develop the present project. The divide-and-conquer strategy and meetings to check our work were key elements in project development. Before the final exposition, we realized that our project was solid and exceeded our expectations. We learned a lot about coding, but more importantly, we learned that union is a strength.

**Conclusion**

Winter is the hardest season of all year. For that reason, we chose the present project. The main challenge was to create a real-world application using all the material learned in the term. It was an excellent exercise not only to show our development skills but also to engage with teamwork and create a base app that can address a real situation. The final result is an app that led us not only to use class knowledge but also tools such as GitHub and explore different sources of information to select the more valuable ones and apply them to our code project.

Projects have challenges and the present is not the exception. Before selecting the Streamlit Library, we researched and tested some libraries to choose the proper one. There was a fixed time, so we organized it to schedule test times to get the right one. In addition, we tested some weather APIs to get the best one. All of them have their limitations, but, in the end, we realized that for the present project, we determined that Open-Meteo is the best choice. The data manipulation ran smoothly when we understood the structure of the data, how to manage it using a DataFrame, and the methods to preprocess it before plotting.

The present project is only the beginning of our development journey. Whether expanding the current one or creating a new one in the future, this idea will lead us to improve our skills and to create better solutions in the near future. As students of artificial intelligence, we need to go beyond and understand that this is the first step to creating worthy apps. The continuous learning process does not stop and now we have more tools to improve and apply in future developments as traditional and machine learning solution developers.

**Future Work**

The current API has a wide number of variables to show. The app can be enhanced by analyzing them and creating a dashboard with more variables not only to show to the user, but the user can customize. In addition, the visual interface can be improved using a more powerful visual library to create a more interactive dashboard. Currently, the current solution is running on a local host, the next step is to test it using a real web server to be accessed in any device. Finally, as artificial intelligence students, the next natural step is to gather more information to apply data manipulation techniques and machine learning methods to predict or classify a feature based on the information provided by the API.

# References

|  |  |
| --- | --- |
| [1] | Streamlit Community, "Streamlit documentation," 06 11 2024. [Online]. Available: https://docs.streamlit.io/. [Accessed 26 11 2024]. |
| [2] | GeeksforGeeks, "GeeksforGeeks," 2024 08 30. [Online]. Available: https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/. |
| [3] | James, "Streamlist tutorial," 04 09 2024. [Online]. Available: https://dev.to/jamesbmour/streamlit-part-1-write-and-text-elements-m9i. [Accessed 25 11 2024]. |
| [4] | O.-M. Community, "Open-Meteo," 19 11 2007. [Online]. Available: https://open-meteo.com/. [Accessed 28 11 2024]. |

**Appendices**

The present project uses the libraries streamlit, pandas, matplotlib, and requests libraries. All of them can be installed via pip command in the current Python virtual environment:

pip install streamlit

pip install pandas

pip install matplotlib

pip install requests

To run the project, it is necessary to locate the main.py file in the cmd and in that location execute the following command:

streamlit run main.py

Once executed, the Streamlit server will open a new tab in the default web browser with the application.

In Visual Studio Code, it is only necessary to view a new terminal, locate the main.py file, and execute the same command mentioned above.

It is not necessary to use any extra dataset. The same app allows us to save the API data and upload it to see it via ‘Option # 1: Local CSV’.

To get the information from the API, it is necessary to go to the ‘Option # 2: Online section’ and ‘Select your plot:’. There are two options: specific date or range of dates. For an specific date, the app shows current weather via ‘Today’ button or input a different date via ‘Input a date (YYYY-MM-DD):’ Both examples are showed in the following pictures:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

The second option is Range of dates. It allows you to write specific dates and shows data after pressing Enter in the ‘Input a final date (YYYY-MM-DD)’. The data plotted is max temperature, min temperature, precipitation, and a comparison of max and min temperature. In addition, there is a ‘Save to CSV’ button to save the current data to a CSV file.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

The downloaded CSV data can be loaded using ‘Option # 1: Local CSV’ browsing it from the local file system or drag and drop the file in the ‘Upload CSV file space’. It shows the current data and a plot.

Captura de pantalla de un celular

Descripción generada automáticamente