

Final Project Report: Weather App

Summary of Contributions and Development Process

The goal of this project was to develop a Python application to provide real-time weather information using data from the OpenWeather API. Key contributions involved setting up the API integration, creating a user-friendly graphical user interface (GUI), implementing search functionality, and fine-tuning the forecast display to offer a seamless user experience. The development process was structured to systematically address these components, ensuring the application is both functional and easy to use.

Key Challenges and Solutions

Several challenges were encountered:

- **GUI Layout Issues:** Creating an intuitive and aesthetically pleasing interface using Tkinter proved challenging. To address this, continuous testing and adjustments were made to achieve a satisfactory layout.
- **Icon Integration:** Incorporating icons into the GUI was difficult initially. Using the Pillow library facilitated the handling and integration of these icons successfully.

Reflection on Python Experience and Libraries Used

Working on this project provided significant insights into Python's capabilities and the utility of its libraries:

- **Tkinter:** Essential for building the GUI, allowing for the development of a user-friendly interface.
- **Requests:** Used for making HTTP requests to retrieve weather data from the OpenWeather API.
- **Pillow:** Assisted in handling and displaying icons within the application.
- **Datetime:** Utilized for managing and displaying date and time information.

Unexpected Findings and Improvements

During the development process, unexpected findings led to improvements:

- I was able to interact with the website to obtain the weather icons and information, so I was able to optimize my code by using the website directly.

Changes from Original Proposal

The project evolved from the initial proposal due to my limitations as a coder and the limitations of the API. Notable changes included simplifying the GUI design to enhance usability and adding more robust error handling mechanisms to manage potential issues with data retrieval and display. In addition, I was unable to retrieve all the information from the API that I wanted so I had to adjust my final product.