**Project Report: Weather App**

**1. Introduction**

**1.1 Project Overview**

The Weather App project aims to provide users with real-time weather information for a given city. The application utilizes the OpenWeatherMap API to fetch weather data and displays it in a user-friendly graphical interface built using the Tkinter library in Python.

**1.2 Objectives**

* Develop a weather application that allows users to check the current weather conditions of any city.
* Implement a user-friendly graphical interface for ease of interaction.
* Utilize geolocation and timezone information to enhance accuracy.

**2. Project Structure**

**2.1 Source Code**

The source code is organized into three main modules:

* **main.py**: The main entry point of the application.
* **weather\_gui.py**: Contains the Tkinter GUI code for the Weather App.
* **weather\_util.py**: Includes utility functions for fetching weather data and handling time-related operations.

**2.2 Image Assets**

The **images/** directory contains image files used in the application, including icons and logos.

**3. Functionality**

**3.1 User Interface**

The application provides a user-friendly interface with the following key features:

* A search box for entering the desired city.
* Display of current weather information, including temperature, condition, wind speed, humidity, description, and pressure.
* Automatic updating of the local time based on the selected city's timezone.

**3.2 Weather Data**

The application fetches weather data from the OpenWeatherMap API, including temperature, weather conditions, wind speed, humidity, and pressure.

**3.3 Error Handling**

The application includes error handling to notify users of invalid entries or other issues during data retrieval.

**4. Dependencies**

The project relies on the following external libraries:

* Tkinter
* Geopy
* Timezonefinder
* Requests
* Pytz

**5. Installation Instructions**

To run the Weather App, follow these steps:

1. Install the required dependencies using **pip install -r requirements.txt**.
2. Execute **main.py** to launch the application.

**6. Conclusion**

The Weather App project successfully achieves its objectives by providing users with a simple and intuitive tool for checking current weather conditions. The use of geolocation and timezone information enhances the accuracy and relevance of the displayed data.

**7. Future Enhancements**

Future development can focus on the following enhancements:

* Incorporating a multi-day weather forecast.
* Adding support for more detailed weather information.
* Enhancing the GUI for a more visually appealing design.

**8. Acknowledgments**

Special thanks to the developers of the libraries and APIs used in this project.