

Title : Weather App Using Python

SRN.No.	31230035
---------	----------

**VISHWAKARMA UNIVERSITY, PUNE**

University Grants Commission (UGC) Approved State Private University

**Project Title:** Weather App

**Project Objective:.** To develop a simple weather application in Python that retrieves and displays current weather information for any given city using the OpenWeatherMap API.

**Project Outcome:** The user will be able to enter the name of a city and receive real-time weather information, including temperature, pressure, humidity, and a brief weather description.

	Project Description	Marks	CO	BTL
1	<p><u>Definition</u></p> <p>weather application using Tkinter for the user interface. It retrieves current weather data from OpenWeatherMap and hourly forecasts from VisualCrossing. Users can enter a city name, and the app displays key weather details such as temperature, conditions, wind speed, and sunrise/sunset times. It also presents a 24-hour forecast with information for each hour, including temperature and weather conditions. The app includes error handling for invalid city names and API errors, ensuring that users are prompted if a valid city is not provided.</p>			
2	<p><u>Methodology used</u></p> <p>❓ <b>GUI Creation: Tkinter</b></p> <ul style="list-style-type: none"> <li>• <b>Library Used:</b> tkinter</li> <li>• Tkinter is used to create the graphical user interface (GUI) of the weather app, including windows, buttons, labels, and frames.</li> </ul> <p>❓ <b>Logo Handling: Pillow (PIL)</b></p> <ul style="list-style-type: none"> <li>• <b>Library Used:</b> Pillow (PIL)</li> <li>• This library resizes and displays the logo image in the app interface.</li> </ul> <p>❓ <b>Weather Data Retrieval: OpenWeatherMap API</b></p> <ul style="list-style-type: none"> <li>• <b>API Used:</b> OpenWeatherMap (API Key: 8d4e8f807af8f877a9b46931b17a21cc)</li> <li>• The API is used to fetch real-time weather data for the given city, including temperature, weather conditions, wind speed, and sunrise/sunset times.</li> </ul> <p>❓ <b>Timezone Conversion: pytz</b></p> <ul style="list-style-type: none"> <li>• <b>Library Used:</b> pytz</li> <li>• Used to adjust and display the local time based on the city's timezone, as provided by the OpenWeatherMap API.</li> </ul> <p>❓ <b>Hourly Forecast Retrieval: VisualCrossing API</b></p> <ul style="list-style-type: none"> <li>• <b>API Used:</b> VisualCrossing (API Key: ZNYH45SCWGSZHQVLV29WLD4ZCM)</li> <li>• This API is used to fetch a 24-hour weather forecast, displaying temperature and conditions for each hour in a grid layout.</li> </ul> <p>❓ <b>HTTP Requests: requests and urllib</b></p> <ul style="list-style-type: none"> <li>• <b>Libraries Used:</b> requests, urllib</li> </ul>			

	<ul style="list-style-type: none"> <li>requests is used to interact with the OpenWeatherMap API, and urllib is used for fetching data from VisualCrossing API.</li> </ul> <p>❓ <b>Error Handling: Tkinter messagebox</b></p> <ul style="list-style-type: none"> <li><b>Tool Used:</b> messagebox from tkinter</li> <li>Error handling in the app displays warnings or errors to users when a city is not found or when an invalid city name is entered.</li> </ul>			
--	--	--	--	--

	<p><u><b>Resource gathering</b></u></p> <p>The app gathers weather data using two APIs: <b>OpenWeatherMap</b> (API key: 8d4e8f807af8f77a9b46931b17a21cc) for real-time weather data like temperature, wind, and sunrise/sunset times, and <b>VisualCrossing</b> (API key: ZNYH45SCWGSZHQLV29WLD4ZCM) for a 24-hour hourly forecast. <b>requests</b> and <b>urllib</b> libraries are used to fetch the data.</p>			
3	<p><u><b>Implementation</b></u></p> <p>The code implements a weather application using Tkinter for the graphical interface. Users input a city name, triggering requests to the OpenWeatherMap API for current weather data and the VisualCrossing API for a 24-hour forecast. The application displays temperature, weather conditions, wind speed, sunrise, and sunset times, along with a grid of hourly forecasts. It incorporates error handling to manage invalid inputs and uses Pillow to handle the logo image. The app dynamically updates based on user input, providing a comprehensive weather overview.</p>			
4	<p><u><b>Output &amp; evidence</b></u></p> <p>The weather app successfully retrieves and displays real-time weather data and a 24-hour forecast for a user-specified city. Upon entering a valid city name, users receive information such as temperature, weather conditions, wind speed, sunrise/sunset times, and a detailed hourly forecast. The application effectively handles errors, providing user-friendly messages when invalid city names are entered. Evidence of functionality can be observed in the smooth interaction with the APIs, displaying accurate weather data and the responsive GUI created with Tkinter, demonstrating an effective integration of multiple libraries and APIs.</p>			

### Course Outcomes

CO No.	Statement
1	To understand basic concepts in Python
2	To develop applications employing different object-oriented concepts
3	To implement linear algebra functions for arrays using NumPy
4	To demonstrate high-performance data manipulation and analysis using PandasLibrary
5	To plot data visualization using Matplotlib and Seaborn libraries.

### Bloom's Taxonomy Level (BTL)

BTL No.	BTL	Statement
1	Remember	Recall facts and basic concepts
2	Understand	Explain ideas or concepts
3	Apply	Use the information in new situations
4	Analyze	Draw connections among ideas
5	Evaluate	Justify a stand or decision
6	Create	Produce new or original work