

Fig 1 : Home Page

The homepage features a gradient header with a logo and navigation buttons linking to the home and weather pages. The main section includes a centered i2ntroduction with a prominent heading and an image. Buttons have a transparent background and change color on hover. The background image of the main section is centered and sized appropriately. The page is styled using embedded CSS for layout and design.

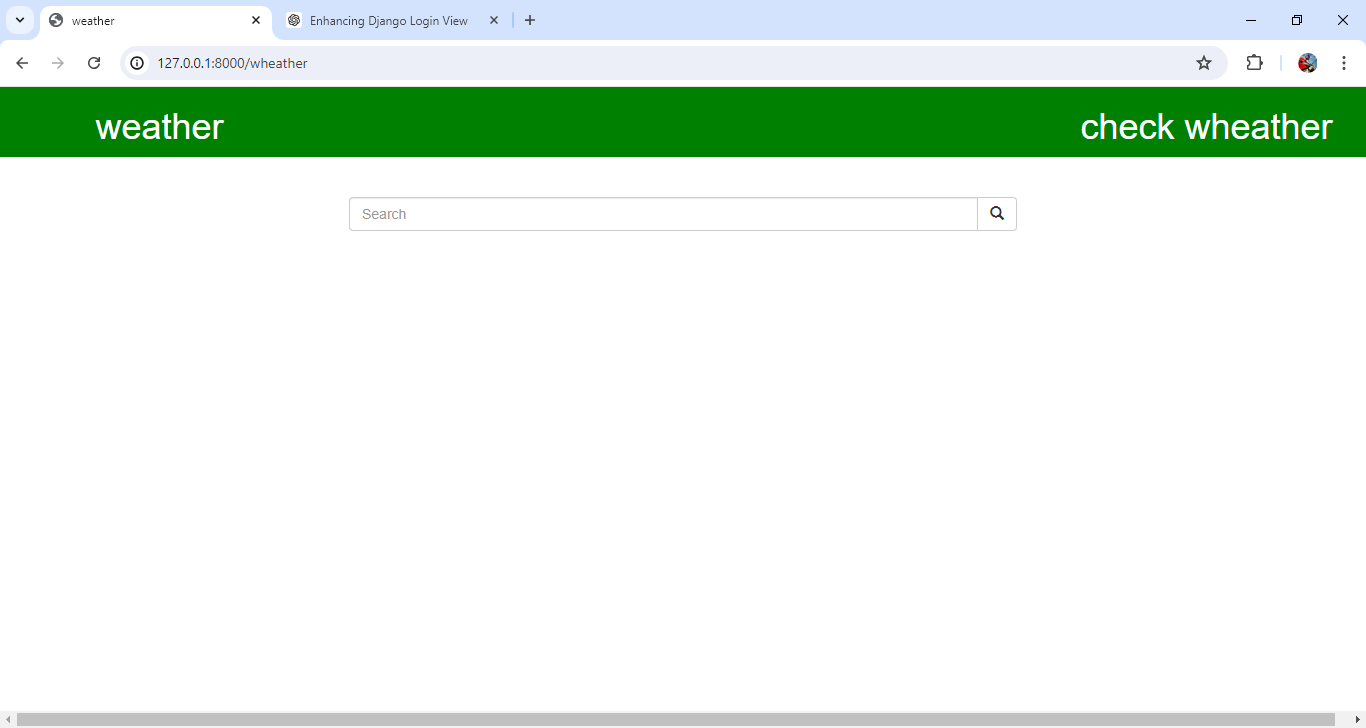
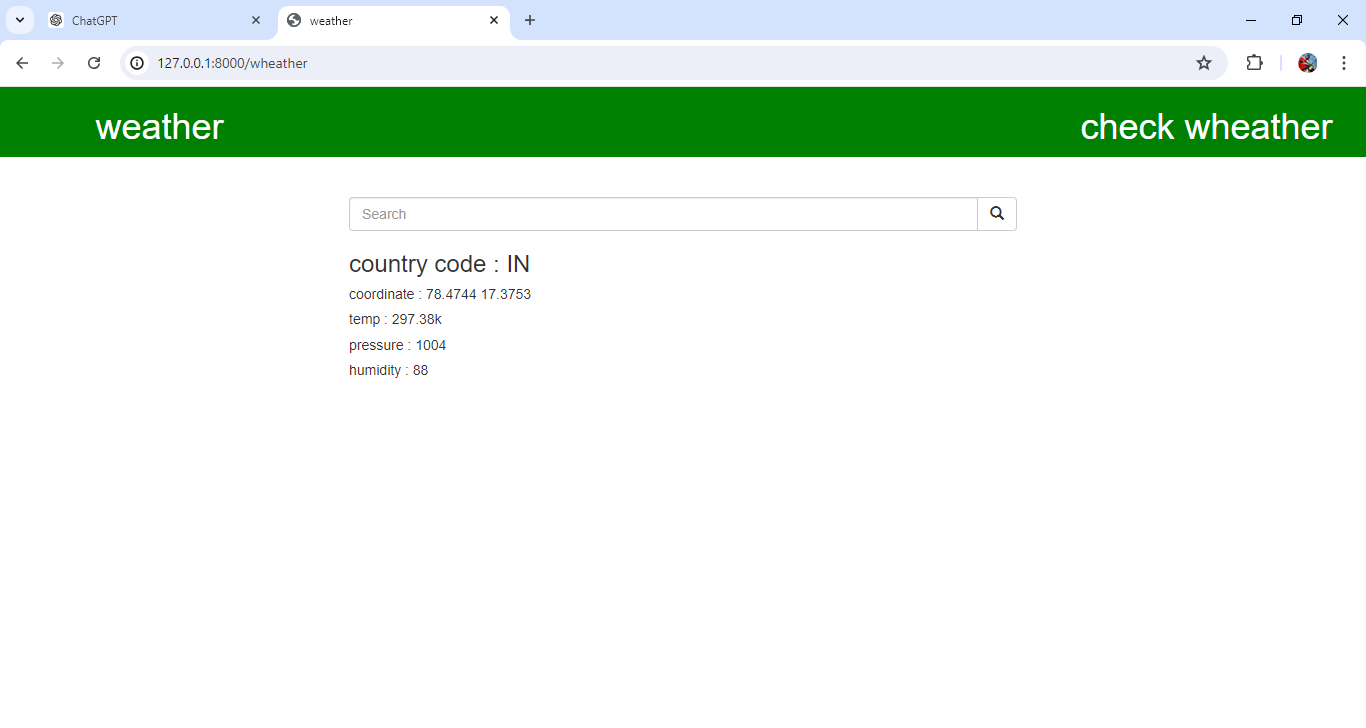


Fig 2: Search Box

The Search function in Wheather Forecasting is designed to handle both GET and POST requests, primarily fetching weather data from the OpenWeatherMap API based on a city name provided by the user. When a POST request is detected, it extracts the city name from the POST data and constructs a URL using an API key to query the OpenWeatherMap API. The JSON response from the API is converted into a Python dictionary, from which relevant weather details such as country code, coordinates, temperature, pressure, and humidity are extracted and prepared for the template. If the request method is not POST, an empty dictionary is passed to the template instead.

Fig 3: Result (output data)

Error handling is robustly implemented through multiple try-except blocks that catch and manage potential errors such as HTTP errors, URL errors, JSON decoding issues, missing keys, and other unforeseen exceptions. These blocks ensure that any errors encountered during the API request or response processing are caught, and appropriate error messages are stored in the data dictionary. Finally, the function renders the weather page template, passing either the weather data or error messages, ensuring the user receives informative feedback regardless of the outcome of the API request.