Weather App Documentation

Project Overview:

The Weather App is a basic web application made with Django. It lets users check the weather and temperature for any city. The app gets weather details from an external API and shows them to the user.

Purpose:

This project is meant to show how to build a simple web app using Django and connect it with an external API to get weather information. It helps in learning how Django works and how to use APIs.

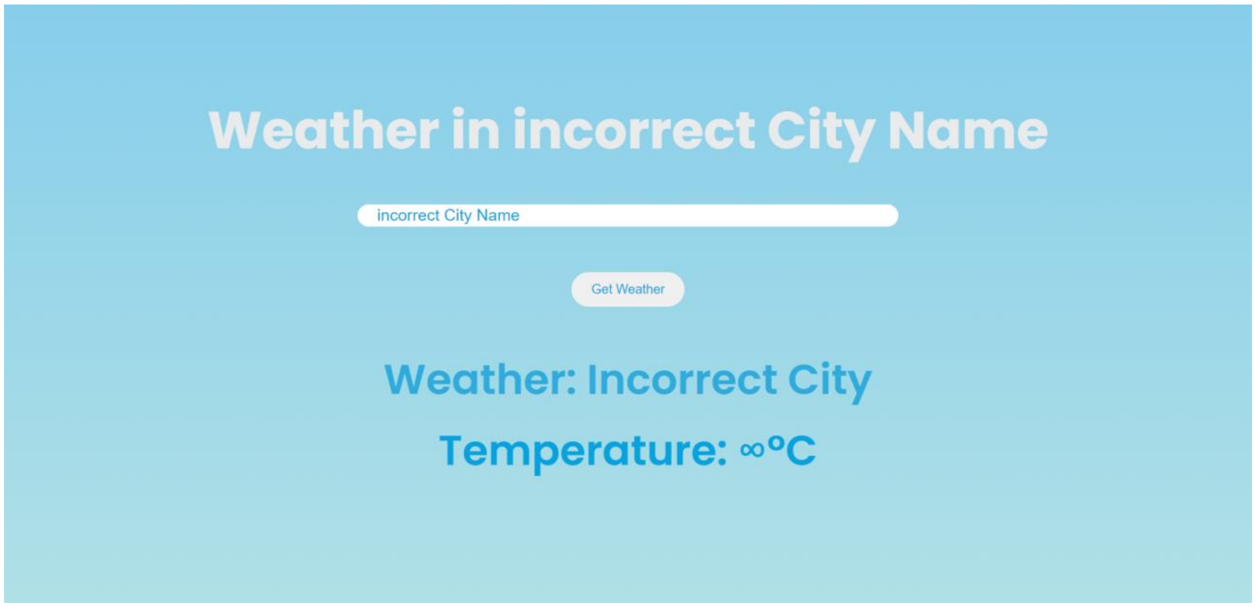
Features:

1. **User Input:**
    - Users type the name of a city into a text box and submit the form.
2. **API Integration:**
    - After the form is submitted, the app asks a weather API for data about the city.
    - The app gets the weather details and shows them on the page.
3. **Display Information:**
    - The app shows the city name, current weather, and temperature in Celsius on the web page.

How to Use:

4. **Running the App:**
    - Make sure Django is installed. Install it with `pip install django` if you haven’t already.
    - Go to your project folder and run `python manage.py runserver`.
    - Open a web browser and go to `http://127.0.0.1:8000/` to see the app.
5. **Checking the Weather:**
    - Type a city name in the input box and click the "Get Weather" button.
    - The page will refresh and show the weather info for that city.

Screenshots:



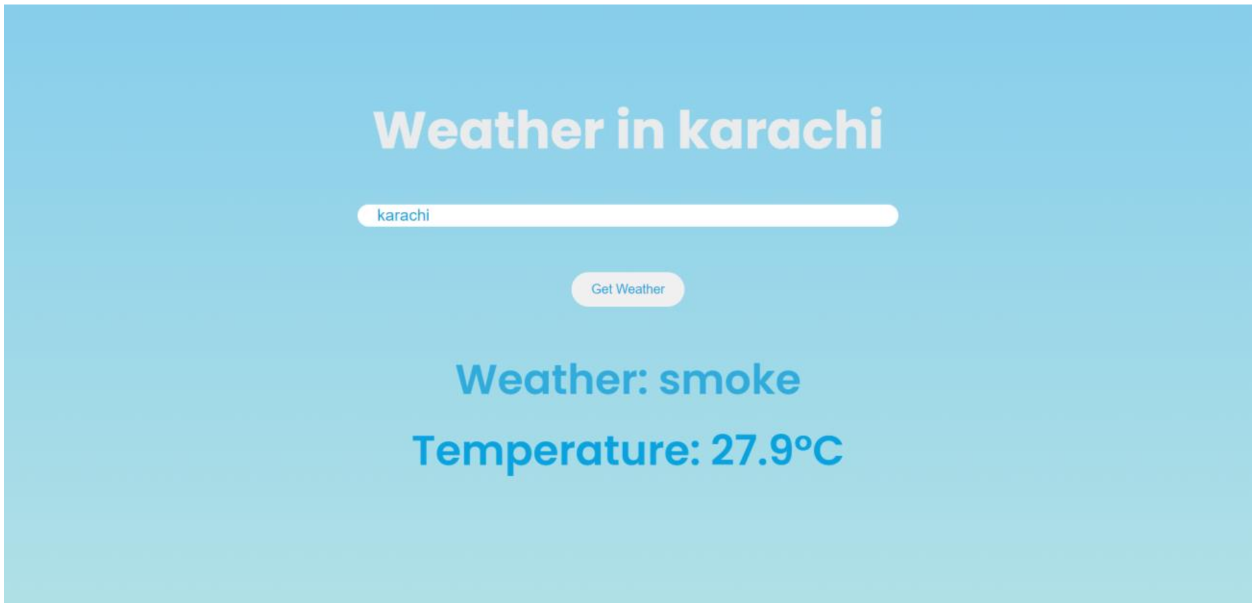
Weather in incorrect City Name

incorrect City Name

Get Weather

Weather: Incorrect City

Temperature: ∞°C



Weather in karachi

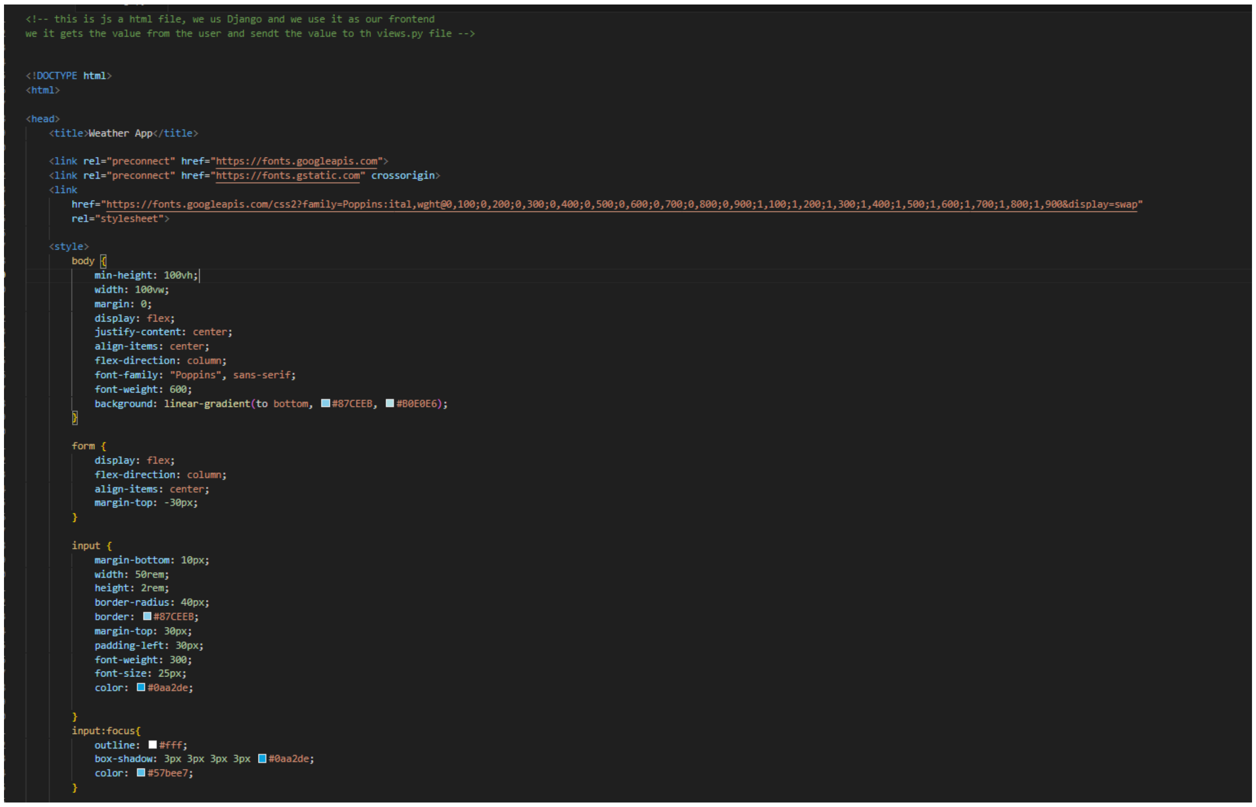
karachi

Get Weather

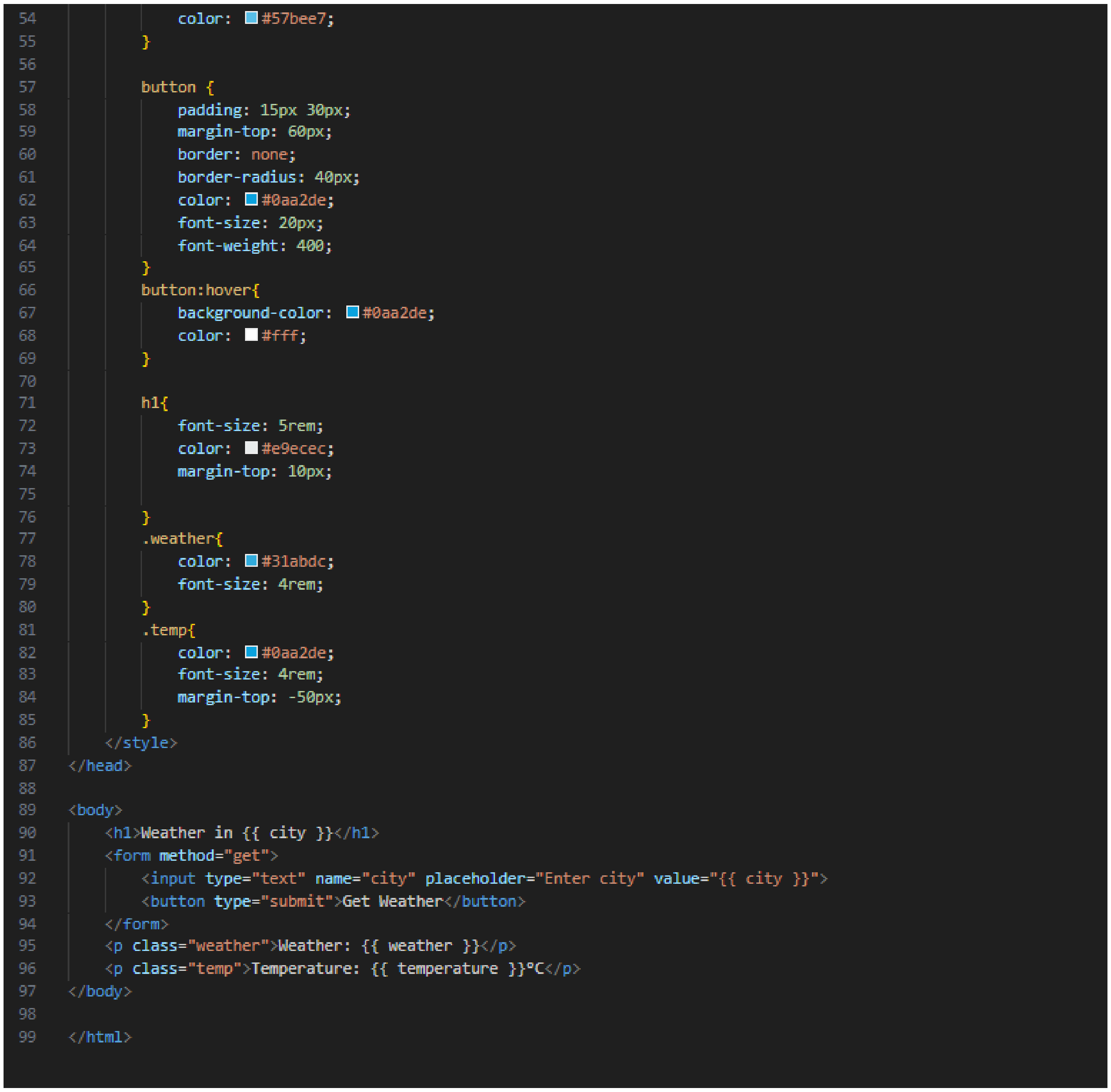
Weather: smoke

Temperature: 27.9°C

Html

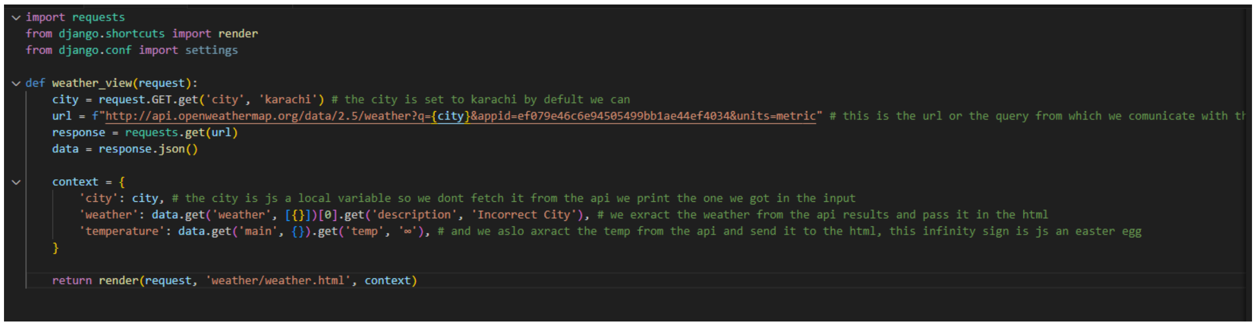


```html
<!-- this is js a html file, we us Django and we use it as our frontend
we it gets the value from the user and sendt the value to th views.py file -->


<!DOCTYPE html>
<html>

<head>
    <title>Weather App</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
        href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap"
        rel="stylesheet">

    <style>
        body {
            min-height: 100vh;
            width: 100vw;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
            font-family: "Poppins", sans-serif;
            font-weight: 600;
            background: linear-gradient(to bottom, #87CEEB, #B0E0E6);
        }

        form {
            display: flex;
            flex-direction: column;
            align-items: center;
            margin-top: -30px;
        }

        input {
            margin-bottom: 10px;
            width: 50rem;
            height: 2rem;
            border-radius: 40px;
            border: #87CEEB;
            margin-top: 30px;
            padding-left: 30px;
            font-weight: 300;
            font-size: 25px;
            color: #0aa2de;

        }
        input:focus{
            outline: #fff;
            box-shadow: 3px 3px 3px 3px #0aa2de;
            color: #57bee7;
        }
```



```
            color: #57bee7;
        }

        button {
            padding: 15px 30px;
            margin-top: 60px;
            border: none;
            border-radius: 40px;
            color: #0aa2de;
            font-size: 20px;
            font-weight: 400;
        }
        button:hover{
            background-color: #0aa2de;
            color: #fff;
        }

        h1{
            font-size: 5rem;
            color: #e9ecec;
            margin-top: 10px;

        }
        .weather{
            color: #31abdc;
            font-size: 4rem;
        }
        .temp{
            color: #0aa2de;
            font-size: 4rem;
            margin-top: -50px;
        }
    </style>
</head>

<body>
    <h1>Weather in {{ city }}</h1>
    <form method="get">
        <input type="text" name="city" placeholder="Enter city" value="{{ city }}">
        <button type="submit">Get Weather</button>
    </form>
    <p class="weather">Weather: {{ weather }}</p>
    <p class="temp">Temperature: {{ temperature }}°C</p>
</body>

</html>
```

Views.py



```
import requests
from django.shortcuts import render
from django.conf import settings

def weather_view(request):
    city = request.GET.get('city', 'karachi') # the city is set to karachi by defult we can
    url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid=ef079e46c6e94505499bb1ae44ef4034&units=metric" # this is the url or the query from which we comunicate with th
    response = requests.get(url)
    data = response.json()

    context = {
        'city': city, # the city is js a local variable so we dont fetch it from the api we print the one we got in the input
        'weather': data.get('weather', [{}])[0].get('description', 'Incorrect City'), # we exract the weather from the api results and pass it in the html
        'temperature': data.get('main', {}).get('temp', '∞'), # and we aslo axract the temp from the api and send it to the html, this infinity sign is js an easter egg
    }

    return render(request, 'weather/weather.html', context)
```

Code Overview:

6. **`views.py`:**
    - Manages getting weather data from the API and sending it to the HTML page.
7. **`urls.py`:**
    - Sets up the website's URL paths and links them to the appropriate functions.
8. **`templates/weather.html`:**
    - Contains the HTML and CSS for how the web page looks.

Conclusion:

The Weather App is a simple project that helps understand the basics of Django and working with APIs. It's a good starting point for learning how to build web apps and handle user input.