

```
pip install requests dash dash-core-components dash-html-components
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.31.0)
Requirement already satisfied: dash in /usr/local/lib/python3.10/dist-packages (2.16.1)
Requirement already satisfied: dash-core-components in /usr/local/lib/python3.10/dist-packages (2.0.0)
Requirement already satisfied: dash-html-components in /usr/local/lib/python3.10/dist-packages (2.0.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests) (2024.2.2)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /usr/local/lib/python3.10/dist-packages (from dash) (2.2.5)
Requirement already satisfied: Werkzeug<3.1 in /usr/local/lib/python3.10/dist-packages (from dash) (3.0.2)
Requirement already satisfied: plotly>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.15.0)
Requirement already satisfied: dash-table==5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.0.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from dash) (7.1.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from dash) (4.11.0)
Requirement already satisfied: retrying in /usr/local/lib/python3.10/dist-packages (from dash) (1.3.4)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from dash) (67.7.2)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (3.1.3)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (2.1.2)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (8.1.7)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (24.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from Werkzeug<3.1->dash) (2.1.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->dash) (3.18.1)
Requirement already satisfied: six>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from retrying->dash) (1.16.0)
```

```
import requests
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output

# Function to fetch weather data from OpenWeatherMap API
def get_weather_data(city):
    api_key = '6bdd8fb1ac7e068de8fd9bd8a437a47b' # Replace with your actual API key
    url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric' # units=metric for Celsius
    response = requests.get(url)
    data = response.json()
    return data

# Create Dash application
app = dash.Dash(__name__)

# Define layout
app.layout = html.Div([
    html.H1("Weather Dashboard", style={'text-align': 'center'}),
    dcc.Input(id='city-input', type='text', placeholder='Enter city name', debounce=True),
    html.Div(id='weather-info', style={'margin-top': '20px'})
])

# Define callback function to update weather information based on user input
@app.callback(
    Output('weather-info', 'children'),
    [Input('city-input', 'value')]
)
def update_weather_info(city):
    if city:
        data = get_weather_data(city)
        if 'main' in data:
            temperature = data['main']['temp']
            weather_description = data['weather'][0]['description'].capitalize()
            humidity = data['main']['humidity']
            wind_speed = data['wind']['speed']
            return html.Div([
                html.H3(f"Weather in {city}:"),
                html.P(f"Temperature: {temperature}°C"),
                html.P(f"Weather Description: {weather_description}"),
                html.P(f"Humidity: {humidity}%"),
                html.P(f"Wind Speed: {wind_speed} m/s")
            ])
        else:
            return html.Div("City not found")

# Run the Dash application
if __name__ == '__main__':
    app.run_server(debug=True)
```

Weather Dashboard

Kansas

Weather in Kansas:

Temperature: 20.54°C

Weather Description: Overcast clouds

Humidity: 82%

Wind Speed: 13.37 m/s