```python
from flask import Flask, jsonify, request
from datetime import datetime
from datetime import timedelta
import requests
import json
import math
import pickle
import pandas as pd


app = Flask(__name__)
# Creating the instance of Flask app

model = pickle.load(open('model.pkl','rb'))
# Loading the ML model into a variable for prediction

def getWeatherData(latitude,longitude):
    now = datetime.now() + timedelta(minutes=330)
    # Server is in UK so setting it to IST
    startTime = now.strftime("%Y-%m-%dT%H:%M:%SZ")
    end = now + timedelta(hours=71)
    # We need 72 hours data so fixing start and end time
    endTime = end.strftime("%Y-%m-%dT%H:%M:%SZ")
    customUrl = 'start_time='+startTime+'&end_time='+endTime+'&lat='+latitude+'&lon='+longitude
    apiUrl = 'https://api.climacell.co/v3/weather/forecast/hourly?unit_system=us&fields=temp,wind_speed,baro_pressure,humidity&apikey=uc5Oou9agPTpbVdeu8bOOK8uuwx5QpzX&'
    # The URL which will retrieve data from Climacell API
    response = requests.get(apiUrl+customUrl)
    # Response stored in a variable
    if response:
        return response.json()      # Returning json format of response


def round_up(n, decimals=0):
    multiplier = 10 ** decimals
    return math.ceil(n * multiplier) / multiplier
    # To achieve values till 4 places after decimal

def tempFunc(data,power):
    # This function will form the object to be sent to front-end
    arrObj = []
    for i,en in enumerate(data):
        jsonObj = {}
        jsonObj['power'] = round_up(power[i],4)
        jsonObj['date'] = en['observation_time']['value'][0:10]
        jsonObj['time'] = en['observation_time']['value'][11:19]
        jsonObj['wind_speed'] = str(en['wind_speed']['value'])
        jsonObj['temperature'] = str(en['temp']['value'])
        jsonObj['humidity'] = str(en['humidity']['value'])
        jsonObj['pressure'] = str(en['baro_pressure']['value'])
        arrObj.append(jsonObj)
    return arrObj;

@app.route('/predict', methods=['GET','POST'])
def predictor():
    if request.method == "GET":
        latitude = request.args.get('latitude')
        longitude = request.args.get('longitude')
    # Receiving latitude and longitude values from front-end

    weatherJson = getWeatherData(latitude,longitude)
    # Storing the weather data in weatherJson variable
    jsonList=[]
    windlist = []
    templist = []
    humidlist = []
    preslist = []
    for ele in weatherJson:
        windlist.append(ele['wind_speed']['value'])
        templist.append(ele['temp']['value'])
        humidlist.append(ele['humidity']['value'])
        preslist.append(ele['baro_pressure']['value'])
    # Creating weather variable list to be fed to model for prediction

    myDict = {
        'Wind_Speed_(miles/h)' : windlist,
            'Temperature' : templist,
            'Humidity' : humidlist,
            'Pressure' : preslist
    }
    inpt_df = pd.DataFrame.from_dict(myDict)
    # DataFrame object which is actually accepted by pickle

    prediction = model.predict(inpt_df)
    # Getting predictions from model and storing it in variable
    modelOutput = tempFunc(weatherJson,prediction)
    # Getting object to be sent to front-end and storing in a variable
    jsonOutput = json.dumps(modelOutput)
    # Converting the object into json format
    return jsonOutput
    # Returning the output to front end

if __name__ == "__main__":
    app.run(debug=True)
```