DATE	28-02-2026
TEAM ID	LTVIP2026TMIDS90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	4 MARKS

6.5 - Application Building:

After the model is built, we will be integrating it to a web application so that normal users can also use it to predict the energy in a no-code manner. In the application, the user provides the required values and get the predictions.

Build the python flask app

In the flask application, the API requests, as well as energy prediction requests, are taken and the results are processed.

Step 1: Import required libraries:

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import requests
```

Step 2: Load the model and initialize flask app:

```
App   =    Flask(__name__)
model = joblib.load('power_prediction_model.pkl')
```

Step 3: Configure app.py for api requests:

Flask file takes the city as input and hits the API to get the weather conditions and send it back to the UI.

```
@app.route('/')
def home():
    return render_template('intro.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')
```

```
@app.route('/windapi', methods=['POST'])
def windapi():
    city = request.form.get('city')
    apikey = 'f54119f50d7337ac8de52db5cc2fbd91'
    url = 'http://api.openweathermap.org/data/2.5/weather?q=' + city +
"&appid=" + apikey
    resp = requests.get(url)
    resp = resp.json()

    temp = str(resp["main"]["temp"]) + "°C"
    humid = str(resp["main"]["humidity"]) + "%"
    pressure = str(resp["main"]["pressure"]) + " mmHg"
    speed = str(resp["wind"]["speed"]) + " m/s"

    return render_template('predict.html', temp=temp, humid=humid,
pressure=pressure, speed=speed)
```

Step 4: Configure the file with predictions:

It takes the inputs from the UI and passes it to the model and sends the predicted output to the UI.

```
@app.route('/y_predict', methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[float(x) for x in request.form.values()]]

    prediction = model.predict(x_test)
    print(prediction)
```

```
    output = prediction[0]

    return render_template('predict.html', prediction_text='The energy
predicted is {:.2f} KWh'.format(output))
```

Step 5: Run the app:

Enter commands as shown below

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