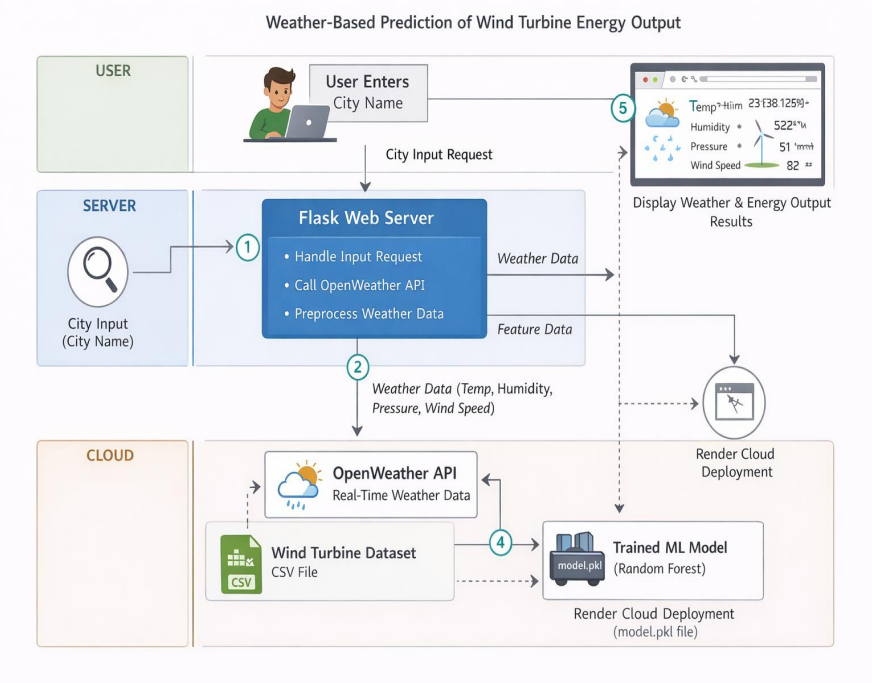
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	15 February 2026
Team ID	LTVIP2026TMIDS80318
Project Name	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
Maximum Marks	4 Marks

Technical Architecture:

The Weather-Based Prediction of Wind Turbine Energy Output architectural diagram as below and the information as per the table1 & table 2

Example: Weather-Based Prediction of Wind Turbine Energy Output



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web page where user enters city name and views weather & prediction results	HTML, CSS
2.	Application Logic-1	Fetch weather data from API and handle user input	Python (Flask)
3.	Application Logic-2	Process weather data and calculate theoretical power	Python
4.	Application Logic-3	Predict wind turbine energy output	Scikit-learn (Machine Learning Model)
5.	Database	Dataset used for training wind energy prediction model	CSV Dataset (Wind Turbine Data)
6.	File Storage	Stores trained model (.pkl file) and dataset	Local File System
7.	External API-1	Fetch real-time weather data (temperature, humidity, pressure, wind speed)	OpenWeather API
8.	Machine Learning Model	Predict wind turbine energy output based on weather conditions	Random Forest Regressor (Scikit-learn)
9.	Infrastructure (Server / Cloud)	Web application deployment	Local System & Render Cloud Deployment

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Web framework and ML libraries used for building and deploying the application	Flask, Scikit-learn, NumPy, Pandas
2.	Security Implementations	API key secured using environment variables; no hardcoding of sensitive data	Environment Variables (OPENWEATHER_API_KEY), HTTPS (Render)
3.	Scalable Architecture	3-Tier architecture: Presentation Layer (UI), Application Layer (Flask), ML Model Layer	HTML/CSS + Flask + Scikit-learn

S.No	Characteristics	Description	Technology
4.	Availability	Cloud deployment ensures 24/7 access to application	Render Cloud Platform
5.	Performance	Lightweight ML model and fast API response for real-time prediction	Random Forest Model, OpenWeather API