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

Date	3 Febraury 2026
Team ID	LTVIP2026TMIDS84120
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 technical architecture integrates weather data sources, wind turbine operational data, and a machine-learning prediction engine within a centralized data processing and storage framework. The system delivers real-time and historical energy output forecasts through a web/mobile dashboard, enabling efficient monitoring and renewable energy management

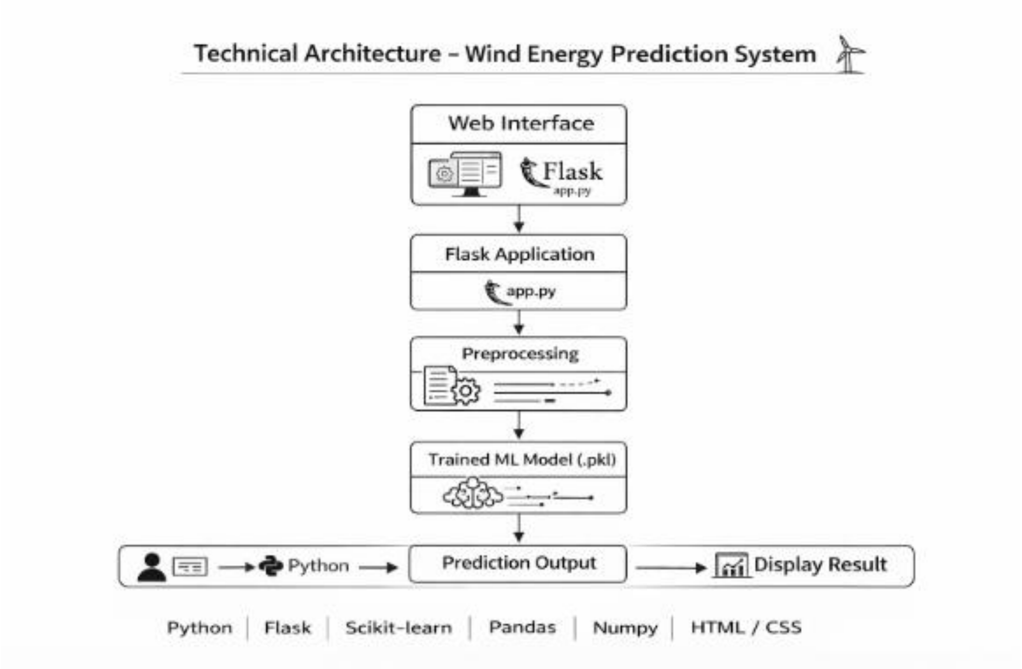

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web interface where user enters weather parameters and views predicted wind energy output	HTML, CSS, JavaScript
2	Application Logic-1	Backend logic to receive user input, preprocess data, and send to ML model for prediction	Python (Flask Framework)
3	Application Logic-2	Data preprocessing logic including missing value handling, encoding, and feature alignment	Python (Pandas, NumPy, Scikit-learn preprocessing)
4	Application Logic-3	Machine learning prediction logic using trained regression model	Python (Scikit-learn Random Forest)
5	Database	Dataset storage containing historical wind turbine sensor and weather data	CSV Files
6	Cloud Database	Optional cloud storage for dataset or model hosting	AWS S3 / Google Drive / IBM Cloud Object Storage
7	File Storage	Storage of trained ML model and feature columns for prediction	Local File System (Joblib .pkl files)
8	External API-1	Weather data source for real-time prediction inputs (optional future integration)	OpenWeatherMap API
9	External API-2	Turbine sensor or SCADA system data integration (optional industrial use)	Wind Farm SCADA API
10	Machine Learning Model	Predict wind turbine power output from weather & sensor parameters	Random Forest Regression
11	Infrastructure (Server / Cloud)	Deployment of Flask web application locally or on cloud server	Local System / Render / AWS / Azure

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	The application uses open-source frameworks for machine learning model development and web deployment	Python, Flask, Scikit-learn, Pandas, NumPy

2	Security Implementations	Input validation is applied on user-entered parameters in the web form; secure model storage and controlled access to prediction endpoints can be implemented during deployment	Flask Input Validation, HTTPS (deployment), OWASP practices
3	Scalable Architecture	The system follows a modular 3-tier architecture with separate layers for user interface, application logic, and machine learning model, allowing independent scaling	Flask Backend + ML Model Service + Web UI
4	Availability	The application can be deployed on cloud platforms ensuring continuous availability and remote access	Local Server / AWS / Azure / Render Cloud
5	Performance	Fast prediction using pre-trained Random Forest model; low-latency inference suitable for real-time prediction requests	Scikit-learn Optimized Model, Joblib Model Loading