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

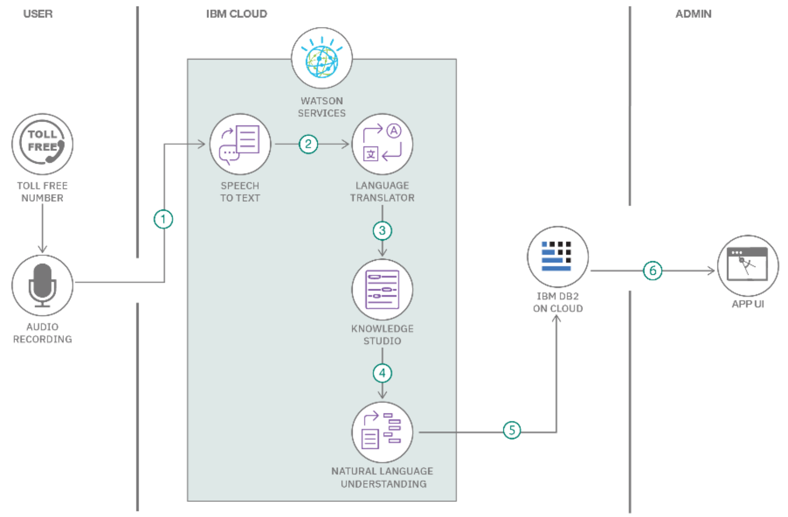
Date	17 February 2026
Team ID	LTVIP2026TMIDS70003
Project Name	Weather-Based Prediction of Wind Turbine Energy
Maximum Marks	4 Marks

Technical Architecture:

The system architecture for the Wind Turbine Energy Prediction project integrates data preprocessing, machine learning model training, external weather API integration, and a Flask-based web application.

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



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 interface for user interaction (intro page, prediction dashboard)	HTML, CSS, JavaScript, Flask Templates
2	Application Logic-1	Handles ML prediction requests (input → model → output)	Python (Flask, Pandas, Scikit-learn)
3	Application Logic-2	Weather data retrieval and processing	Python (Requests library, OpenWeather API)
4	Application Logic-3	Data preprocessing, training, evaluation, and saving model	Python (NumPy, Pandas, Matplotlib, Scikit-learn, Joblib)
5	Database	Stores dataset (historical wind turbine data)	CSV files (local storage)
6	Cloud Database	Optional extension for scalability	AWS RDS / MongoDB Atlas (future scope)
7	File Storage	Stores trained model and datasets	Local filesystem (.sav, .csv)
8	External API-1	Fetches live weather conditions	OpenWeather API
9	External API-2	(Optional future integration) e.g., grid demand API	Energy Grid APIs
10	Machine Learning Model	Predicts wind turbine energy output	Random Forest Regressor
11	Infrastructure	Deployment environment	Local Server (Flask), Cloud-ready (Docker, Kubernetes, AWS/GCP/Azure)

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for ML and web development	Flask, Scikit-learn, Pandas, NumPy, Matplotlib
2	Security Implementations	Basic security for API keys, input validation, and server protection	API key encryption, HTTPS, Flask form validation
3	Scalable Architecture	3-tier architecture (UI → Flask backend → ML model) with cloud deployment	Flask + Docker + Kubernetes (future scope)
4	Availability	Can be hosted on cloud with load balancers for high availability	AWS/GCP/Azure Load Balancers
5	Performance	Optimized ML model, caching weather API responses, lightweight Flask server	Flask caching, CDN (future scope), Random Forest efficiency

---

This design phase shows your project as a professional architecture document:

- **Table-1 captures the components and technologies.**
- **Table-2 highlights application characteristics like scalability, security, and performance.**