WEATHER-BASED PREDICTION OF WIND TURBINE ENERGY OUTPUT: A NEXT-GENERATION APPROACH TO RENEWABLE ENERGY MANAGEMENT

Chapter - 1
Introduction

1.1 - Project Overview:

Renewable energy plays a crucial role in solving major global challenges such as climate change, rapidly growing energy demands, and the decreasing availability of fossil fuels. Among the different types of renewable energy, wind energy is especially important because it is widely available, environmentally friendly, and capable of producing large amounts of electricity.

This project focuses on developing a **Weather-Based Prediction System for Wind Turbine Energy Output** using advanced data analytics and machine learning. By analyzing historical weather data along with turbine performance records, the system identifies patterns and relationships between climate conditions and power generation levels.

Different algorithms such as Linear Regression, Random Forest, Support Vector Machines, and Neural Networks may be tested to identify the most accurate prediction model. The model's performance will be evaluated using accuracy scores and standard error measurements like MSE and RMSE. Once the best model is selected, it will be integrated into a working system capable of predicting wind energy output using real-time or forecasted weather data.

The system is designed to work in several stages. First, important meteorological and wind turbine datasets are collected, including information such as wind speed, temperature, humidity, pressure levels, turbine rotation speed, and previous energy output. The collected data then goes through preprocessing to remove errors, handle missing values, and convert it into a suitable format for analysis. After this, machine learning models are trained to understand how different weather factors influence turbine performance.

In summary, **"Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management"** provides a smart and practical solution to one of the major challenges in wind power systems. By predicting energy output more accurately based on weather conditions, the project helps improve efficiency, maintain stable power supply, and support the global shift toward sustainable and eco-friendly energy systems.

1.2 - Problem Specification:

Wind energy is one of the most promising renewable energy sources, but its power generation capability is highly uncertain because it depends entirely on changing weather conditions. Factors such as wind speed, temperature, humidity, and atmospheric pressure continuously fluctuate, making it difficult to accurately predict how much electricity a wind turbine will generate at any given time. Due to this uncertainty, power grid operators, energy planners, and wind farm managers face major challenges in forecasting energy supply, planning power distribution, and maintaining grid stability.

Therefore, there is a **strong need for an intelligent and accurate weather-based wind energy prediction system**. The system must be capable of analysing multiple weather parameters, learning from historical data, handling dynamic environmental variations, and providing reliable predictions of wind turbine energy output. Such a system will help ensure better power planning, improved energy management, cost efficiency, and enhanced stability in renewable energy supply.

1.3 - Purpose & Objectives of the Project:

This project aims to develop an intelligent system that could forecast the generated energy output of wind turbines, given ever-changing weather conditions. Because wind power generation greatly relies on environmental factors such as wind speed, temperature, humidity, and atmospheric pressure, it would be very beneficial to have appropriate forecasting to reduce uncertainties in renewable energy production. It will contribute to better planning of power, higher grid stability, improved resource management, and efficient utilization of wind energy in real-world applications.

Objectives of the Project:

1.To analyse the impact of various weather parameters on wind turbine performance.

2.To collect and prepare relevant datasets for accurate prediction.

3.To design and develop a machine learning-based prediction model.

4.To evaluate and enhance the accuracy of the prediction model.

5.To build a system capable of providing real-time or forecast-based wind energy predictions.

6.To contribute to smart and sustainable energy management.

1.4 - Benefits of the Project:

1. Accurate Energy Prediction.
2. Better Power Planning and Grid Management.
3. Efficient Utilization of Wind Resources.
4. Reduced Operational and Maintenance Costs.
5. Improved Decision-Making.
6. Supports Sustainable and Green Energy Goals.
7. Enhanced Reliability of Renewable Energy Systems.

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	2 MARKS

Chapter-2
Indentation Phase

2.1 - Problem Statement:

The increasing global demand for clean and sustainable energy has led to rapid growth in wind power generation. Wind energy is one of the most promising renewable energy sources; however, its output is highly variable and depends strongly on weather conditions such as wind speed, wind direction, temperature, air pressure, and humidity. This variability makes it difficult for power grid operators and energy planners to accurately predict the amount of energy that will be generated at any given time.

Traditional methods of estimating wind turbine power output often rely on simplified models or fixed power curves, which do not fully capture the complex and dynamic relationship between weather parameters and actual energy production. As a result, these methods can lead to inaccurate forecasts, inefficient energy management, poor load balancing, and increased operational costs.

Inaccurate prediction of wind energy output also affects grid stability, scheduling of backup power sources, and overall reliability of renewable energy integration into the power system. With the growing penetration of wind energy in modern power grids, there is a strong need for a more intelligent, data-driven, and accurate prediction system.

Therefore, the problem addressed in this project is the development of a weather-based predictive model that can accurately estimate wind turbine energy output using historical weather and turbine performance data. The goal is to improve forecasting accuracy, support better decision-making in energy management, and enhance the efficiency and reliability of wind power generation systems.

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

2.2 - Empathy Map Canvas:

The Empathy Map Canvas is used to understand the needs, challenges, and expectations of the stakeholders involved in wind energy generation and management. In this project, the primary users of the proposed system include wind farm operators, energy planners, grid managers, and maintenance engineers. The empathy map helps in designing a system that is user-centric, practical, and aligned with real-world operational requirements.

1. Thinks:

- Needs accurate and reliable predictions of wind turbine energy output.
- Thinks about how to reduce power fluctuations and improve grid stability.
- Worries about unexpected drops or spikes in power generation due to changing weather.
- Considers how to optimize turbine performance and maintenance schedules.
- Wants to make better decisions using data instead of guesswork.

2. Feels:

- Feels pressure to maintain consistent and reliable power supply.
- Feels concerned about losses caused by poor forecasting and inefficient planning.
- Feels responsible for the safety, performance, and efficiency of wind turbines.
- Feels motivated to adopt smarter, technology-driven solutions.
- Feels frustrated when traditional prediction methods fail to give accurate results.

3. Says:

- “We need more accurate power output forecasts.”
- “Weather changes are making energy planning difficult.”
- “Unexpected power variation affects grid management.”
- “We need a system that can help us plan better and reduce risks.”

- "Data-driven tools can improve our decision-making."

4. Does:

- Monitors weather data and turbine performance regularly.
- Uses historical data and basic models to estimate power output.
- Plans energy distribution and backup power based on forecasts.
- Schedules maintenance based on performance trends and issues.
- Analyses reports and dashboards to track energy production.

5. Pain Points:

- Inaccurate or unreliable power output predictions.
- Difficulty in handling sudden changes in weather conditions.
- Risk of energy loss, grid imbalance, and operational inefficiency.
- Dependence on traditional models that do not adapt well to real-world data.
- Increased operational costs due to poor planning and forecasting errors.

6. Gains:

- More accurate and reliable energy output predictions.
- Better planning and management of power generation and distribution.
- Improved turbine performance and reduced downtime.
- Enhanced decision-making using data-driven insights.
- Increased efficiency, reliability, and sustainability of wind energy systems.

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

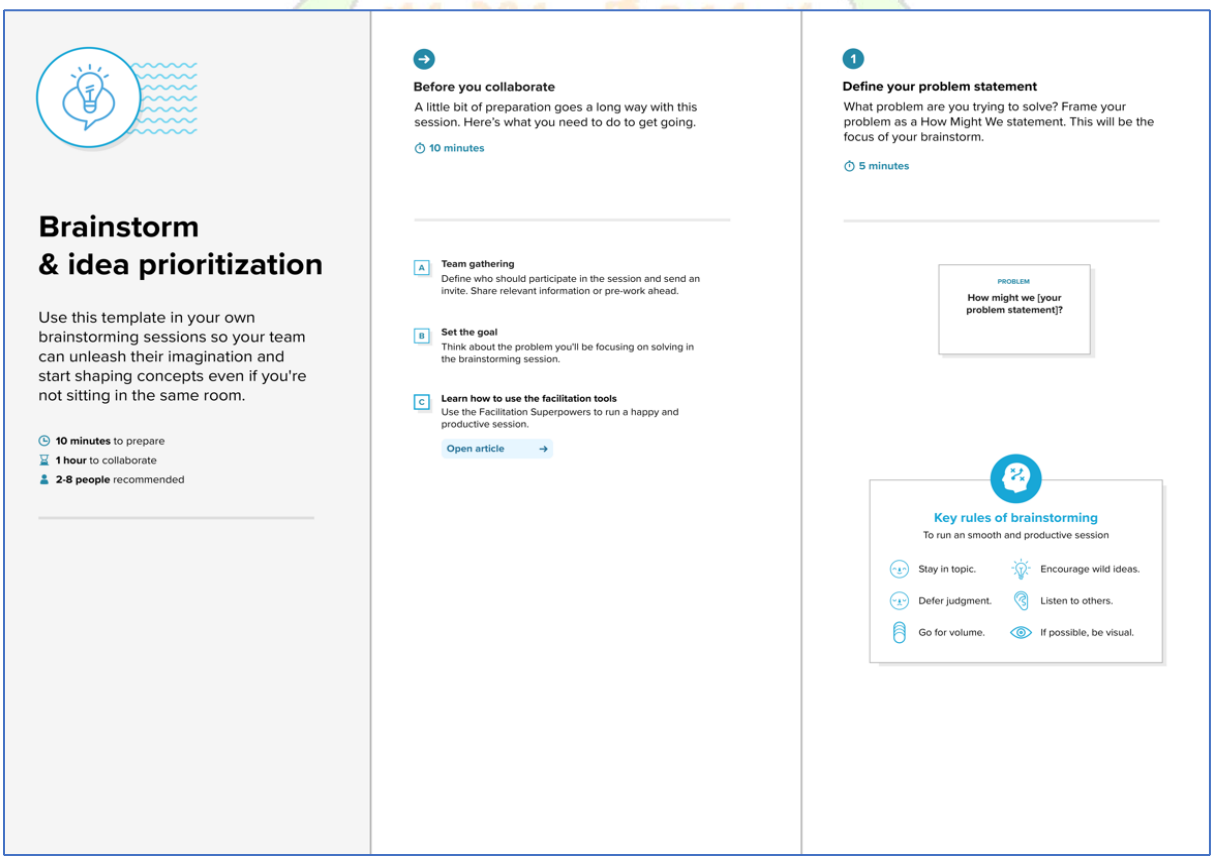
2.3 – Brainstroming:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

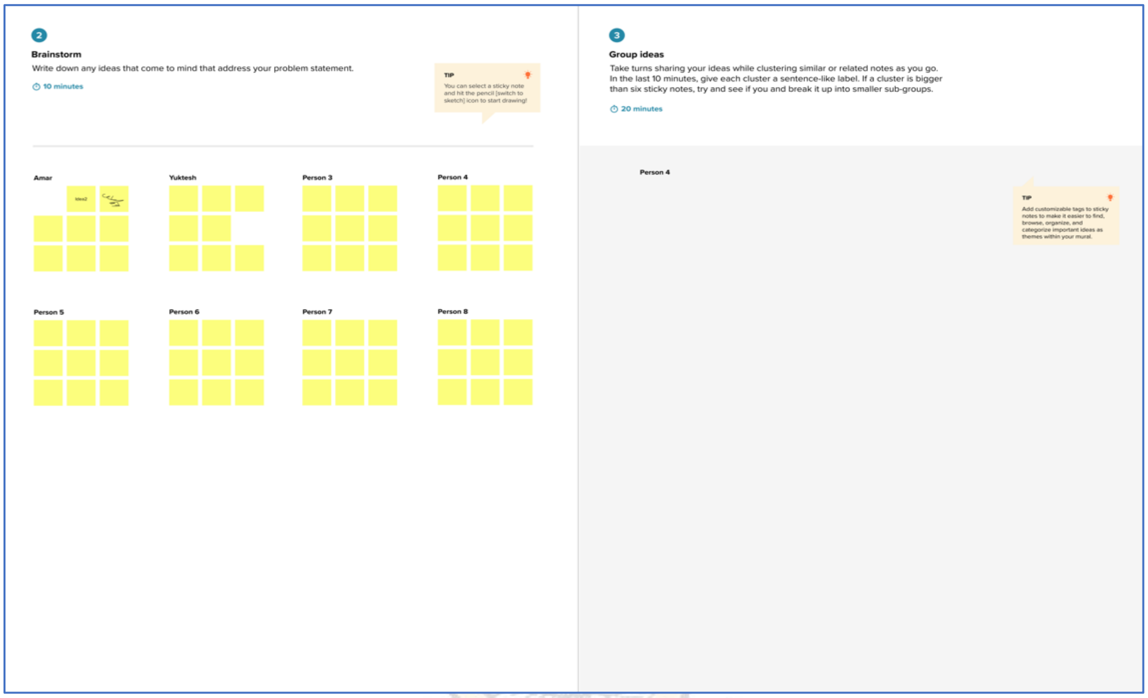
Reference: https://www.mural.co/templates/brainstorm-and-idea-prioritization

Step-1: Team Gathering, Collaboration and Select the Problem Statement:



Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

Step-2: Brainstorm, Idea Listing and Grouping:

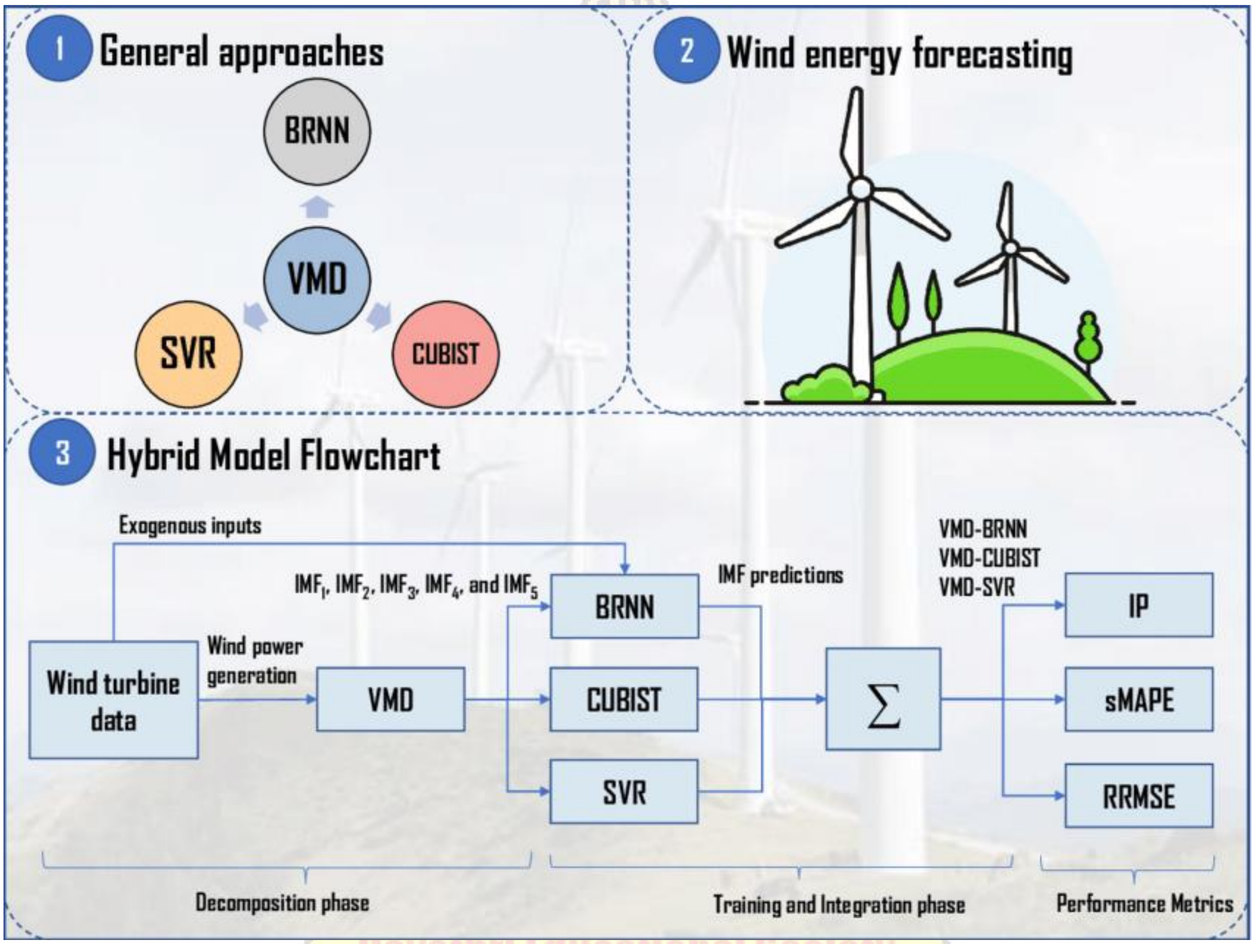


Step-3: Idea Prioritization

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	2 MARKS

Chapter-3
Requirement Analysis

3.1 - Customer Journey map:



Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	2 MARKS

3.2 - Solution Requirement:

The proposed Weather-Based Prediction of Wind Turbine Energy Output System aims to provide an accurate, reliable, and user-friendly solution for forecasting power generation using weather data and historical turbine performance. The solution must satisfy both functional and non-functional requirements to ensure effective operation, usability, and performance.

3.2.1 Functional Requirements:

The system should be able to:

1. Accept and store historical weather data and wind turbine performance data.
2. Preprocess data by handling missing values, removing noise, and normalizing inputs.
3. Analyse key weather parameters such as wind speed, wind direction, temperature, pressure, and humidity.
4. Train machine learning models using historical data for power output prediction.
5. Generate accurate predictions of wind turbine energy output for given weather conditions.

3.2.2 Non-Functional Requirements:

The system should:

1. Be easy to use with a simple and intuitive user interface.
2. Provide fast response time for data processing and prediction generation.
3. Ensure accuracy and reliability of predictions.
4. Be scalable to handle large datasets from multiple wind turbines.
5. Maintain data security and integrity.

3.2.3 Hardware Requirements:

- Processor: Intel i3 or higher
- RAM: Minimum 8 GB
- Storage: Minimum 256 GB HDD/SSD
- System: Standard desktop or laptop computer

3.2.4 Software Requirements:

- Operating System: Windows / Linux
- Programming Language: Python
- Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn (optional), Flask/Streamlet (for UI, if used)
- Development Environment: Anaconda / Jupiter Notebook / VS Code
- Database (Optional): MySQL / SQLite / CSV files

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	3 MARKS

3.3 - Data Flow Diagram:

```
+-------------------------+
|  Weather Data Sources   |
| (API / Sensors / CSV)   |
+-----------+-------------+
            |
            | Raw Weather Data
            v
+-------------------------+
|   Data Collection &     |
|   Integration Module    |
+-----------+-------------+
            |
            | Cleaned Data
            v
+-------------------------+
|  Data Preprocessing     |
|   - Missing Value Fix   |
|   - Normalization       |
|   - Feature Engineering |
+-----------+-------------+
            |
```

```
           | Processed Features
                   v
      +-----------------------+
      |  Machine Learning     |
      |  Model Module         |
      |  (Random Forest /     |
      |   XGBoost / LSTM)     |
      +-----------+-----------+
                  |
        | Predicted Wind Power
                  v
      +-----------------------+
      |  Performance Evaluation|
      |  (RMSE, MAE, R²)      |
      +-----------+-----------+
                  |
            | Final Output
                  v
      +-----------------------+
      |  Energy Management    |
      |  Dashboard / Grid     |
      |  Optimization System  |
      +-----------------------+
```

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	3 MARKS

3.4 - Technology Stack:

The technology stack defines the set of tools, programming languages, frameworks, and platforms used to design, develop, and deploy the Weather-Based Prediction of Wind Turbine Energy Output System. The selected technologies ensure accuracy, scalability, ease of development, and efficient performance of the system.

1. Programming Language

- **Python**

  Python is used as the primary programming language due to its simplicity, readability, and strong support for data science and machine learning applications. It provides a rich ecosystem of libraries for data processing, analysis, visualization, and model development.

2. Data Handling and Processing

- **Pandas –** Used for data loading, cleaning, preprocessing, and manipulation.
- **NumPy –** Used for numerical computations and handling multi-dimensional arrays.

3. Machine Learning Libraries

- **Scikit-learn –** Used for building, training, testing, and evaluating machine learning models such as regression models, decision trees, and ensemble methods.
- **(Optional) TensorFlow / PyTorch –** Can be used for advanced deep learning models if required for higher accuracy.

4. Data Visualization

- **Matplotlib –** Used to create basic plots and graphs for data analysis and result visualization.
- **Seaborn –** Used for advanced and attractive statistical visualizations (optional).

5. Development Environment

- **Anaconda –** Used for managing Python environments and packages.

- **Jupyter Notebook** – Used for experimentation, data analysis, and model development.
- **VS Code / PyCharm** – Used for writing and managing project code.

6. User Interface / Web Framework (Optional)

- **Flask / Streamlet** – Used to build a simple web-based interface or dashboard to display predictions, charts, and reports in a user-friendly manner.

7. Data Storage

- **CSV / Excel Files** – Used for storing datasets during development and testing.
- **SQLite / MySQL (Optional)** – Can be used for structured storage if the system is extended to a larger scale.

8. Version Control

- **Git / GitHub** – Used for source code management, version control, and collaboration.

9. Operating System

- **Windows / Linux** – The system can be developed and executed on either platform.

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	5 MARKS

Chapter 4

Project Design

4.1 - Problem Solution Fit:

The problem identified in this project is the difficulty of accurately predicting wind turbine energy output due to the highly variable and unpredictable nature of weather conditions. Traditional forecasting methods and static power curve models often fail to capture the complex relationship between multiple weather parameters and actual power generation, leading to inaccurate predictions, inefficient energy planning, and challenges in grid management.

The proposed Weather-Based Prediction of Wind Turbine Energy Output System directly addresses these challenges by using a data-driven, machine learning-based approach. The solution integrates historical weather data and turbine performance data to learn patterns and relationships between input weather variables (such as wind speed, wind direction, temperature, pressure, and humidity) and the resulting energy output.

This approach provides a better fit to the problem because:

1. **Handles Weather Variability:**
   The model adapts to changing weather patterns by learning from real historical data instead of relying on fixed mathematical formulas.
2. **Improves Prediction Accuracy:**
   Machine learning algorithms can capture non-linear and complex relationships between inputs and output, resulting in more accurate and reliable energy output forecasts.
3. **Supports Better Decision-Making:**
   Accurate predictions help operators plan power distribution, manage grid stability, and schedule maintenance more effectively.

4. **Reduces Operational Risk:**

   By forecasting potential drops or peaks in power generation, the system helps reduce uncertainty and operational losses.

5. **Scalable and Flexible:**

   The solution can be extended to multiple turbines, larger datasets, and additional weather parameters without major changes to the system design.

6. **User-Friendly and Practical:**

   With dashboards and visual reports, the system presents complex predictions in a simple and understandable form for end users.

   Thus, the proposed solution fits the problem well by transforming unreliable, traditional forecasting methods into an intelligent, accurate, and practical prediction system, improving the efficiency, reliability, and sustainability of wind energy management.

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	5 MARKS

4.2 - Proposed Solution:

The proposed solution is a Weather-Based Wind Turbine Energy Output Prediction System that uses historical weather data and turbine performance data to accurately forecast power generation. The system applies machine learning techniques to model the complex relationship between weather conditions and energy output, enabling more reliable and data-driven predictions compared to traditional methods.

Key Components of the Proposed Solution

1. **Data Collection:**
   The system collects historical data such as wind speed, wind direction, temperature, air pressure, humidity, and actual turbine power output. This data forms the foundation for training and testing the prediction models.
2. **Data Preprocessing:**
   The collected data is cleaned and prepared by handling missing values, removing noise, and normalizing features. This step ensures high-quality input data and improves the accuracy of the prediction model.
3. **Feature Analysis and Selection:**
   Important weather parameters that influence power generation are identified and selected. This helps the model focus on the most relevant inputs affecting wind turbine performance.
4. **Model Training Using Machine Learning:**
   Machine learning algorithms (such as regression models, decision trees, or ensemble methods) are trained using historical data. These models learn the relationship between weather conditions and energy output and can generalize to new, unseen data.
5. **Prediction and Forecasting:**

Once trained, the model predicts future wind turbine energy output based on new or forecasted weather data. The system can provide short-term or long-term predictions depending on user requirements.

6. **Visualization and Reporting:**

   The predicted results are displayed using charts, graphs, and dashboards. The system also allows comparison between predicted and actual output to evaluate performance and improve model accuracy.

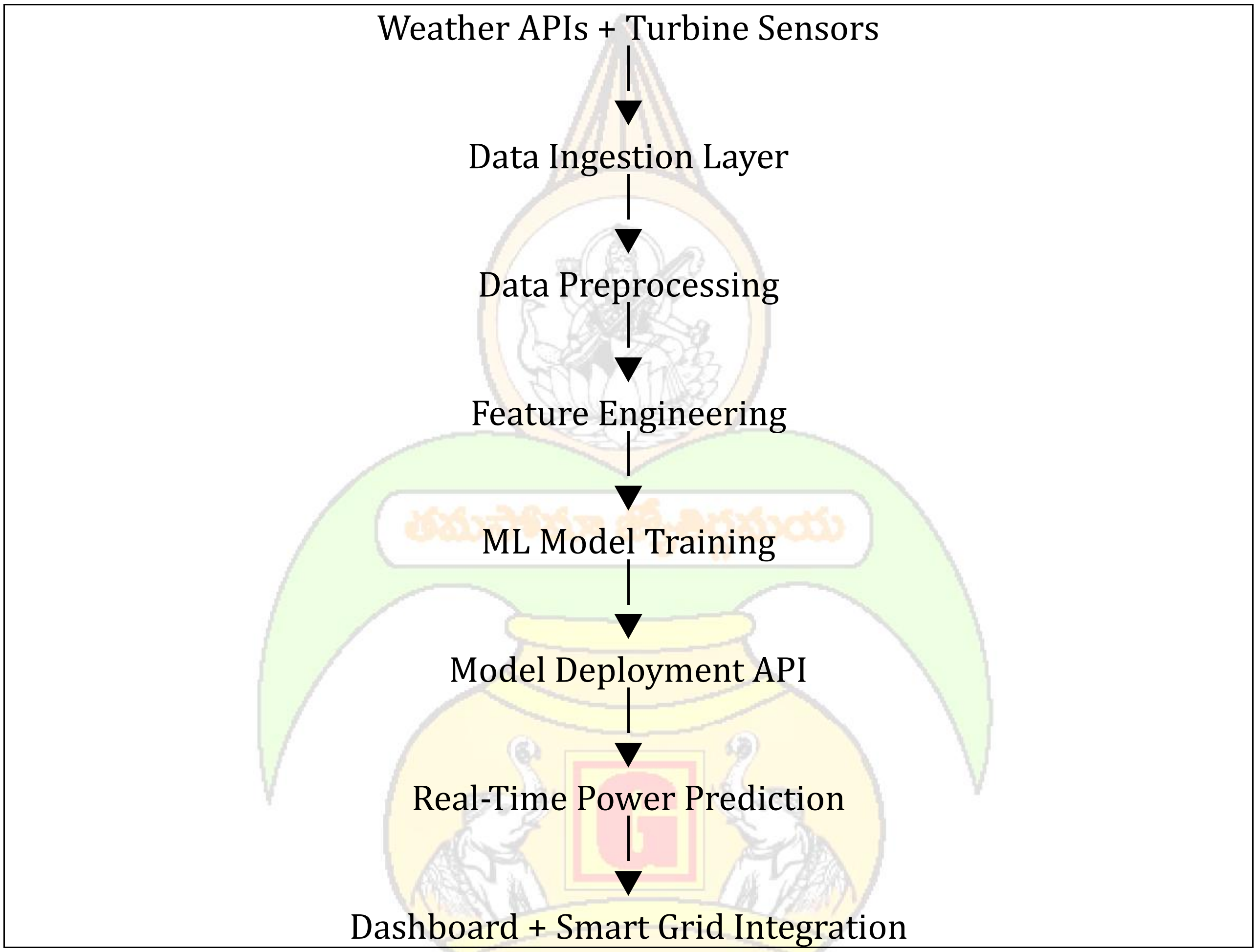
Benefits of the Proposed Solution

- Improved accuracy in wind energy output prediction
- Better utilization of renewable energy resources
- Enhanced decision-making for energy management
- Reduced uncertainty caused by weather variability
- Increased efficiency and reliability of wind power systems

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	5 MARKS

4.3 - Solution Architecture:



Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	1 MARK

Chapter 5

Project Planning and Scheduling

5.1 - Project Milestones & Tasks:

1 Data Collection

Objective: Gather high-quality, relevant data required for the project.

Tasks:

- Identify reliable data sources (databases, APIs, web scraping, surveys, etc.)
- Collect structured and unstructured data
- Ensure data relevance and completeness
- Store data in a centralized database or storage system
- Maintain data documentation for reference

Deliverables:

- Raw dataset
- Data source documentation
- Data storage setup

2 Data Pre-Processing

Objective: Clean and prepare the data for model training and analysis.

Tasks:

- Remove duplicates and handle missing values
- Handle outliers and inconsistent data
- Data normalization or scaling

- Feature engineering and selection
- Encode categorical variables
- Split dataset into training and testing sets

Deliverables:

- Cleaned dataset
- Feature-engineered dataset
- Preprocessing scripts

3 Model Building

Objective: Develop and train a machine learning model.

Tasks:

- Select appropriate algorithm(s)
- Train model using training dataset
- Hyperparameter tuning
- Model validation and evaluation
- Compare performance metrics
- Finalize best-performing model

Deliverables:

- Trained model
- Evaluation report (Accuracy, Precision, Recall, F1-score, etc.)
- Saved model file

4 API Integration

Objective: Expose the trained model via an API for external access.

Tasks:

- Develop REST API endpoints
- Integrate model with backend framework (Flask / FastAPI / Django)

- Implement request validation
- Add error handling
- Test API responses
- Secure API (authentication if required)

Deliverables:

- Functional API
- API documentation (Swagger/Postman collection)
- Deployment-ready backend

5 Web Integration

Objective: Integrate API into a web-based user interface.

Tasks:

- Design user-friendly frontend interface
- Connect frontend to backend API
- Display model predictions dynamically
- Implement input validation
- Perform end-to-end testing
- Deploy web application

Deliverables:

- Fully functional web application
- User interface design
- Live deployment link (if applicable)

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

DATE	28-02-2026
TEAM ID	LTVIP2026TMID90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	2 MARKS

5.2 - Sprint Delivery Plan:

◆ Phase 1: Live Sessions (Week 1–6)

Objective: Build strong foundational knowledge and prepare interns for real-time project development.

🟦 Sprint 1 (Week 1–2): Fundamentals & Tools

Focus Areas:

- Introduction to Internship Program
- Programming Fundamentals (Python / Relevant Tech Stack)
- Git & GitHub
- Development Environment Setup
- Basics of Databases

Deliverables:

- Setup development environment
- GitHub repository creation
- Mini practice assignments

🟦 Sprint 2 (Week 3–4): Data & Backend Foundations

Focus Areas:

- Data Handling (Pandas / Data Structures)
- Data Cleaning Techniques
- Introduction to APIs
- Backend Basics (Flask / FastAPI)
- SQL & Database Integration

Deliverables:

- Data preprocessing assignment
- Basic API development task
- Database connectivity demo

🟦 Sprint 3 (Week 5–6): Machine Learning & Deployment Basics

Focus Areas:

- Machine Learning Fundamentals
- Model Training & Evaluation
- REST API Integration with Model
- Introduction to Web Integration
- Deployment Overview

Deliverables:

- Simple ML Model
- Model evaluation report
- API with working prediction endpoint

🔹 Phase 2: Project Work (Week 7–15)

Objective: Apply learned skills to build a complete end-to-end project.

🟩 Sprint 4 (Week 7–8): Project Planning & Data Collection

Activities:

- Finalize project topic
- Define problem statement
- Collect dataset
- Perform initial data analysis (EDA)

Deliverables:

- Project proposal document

- Dataset documentation
- EDA report

🟩 **Sprint 5 (Week 9–10):** Data Preprocessing & Feature Engineering

Activities:

- Clean dataset
- Handle missing values & outliers
- Feature engineering
- Data transformation
- Train-test split

Deliverables:

- Cleaned dataset
- Preprocessing pipeline
- Feature documentation

🟩 **Sprint 6 (Week 11–12):** Model Development & Optimization

Activities:

- Train multiple models
- Hyperparameter tuning
- Model comparison
- Performance evaluation

Deliverables:

- Best performing model
- Evaluation metrics report
- Saved model artifact

🟩 **Sprint 7 (Week 13–14):** API Development & Integration

Activities:

- Develop REST API
- Integrate trained model
- Implement validation & error handling
- Test API endpoints

Deliverables:

- Functional API
- API documentation
- Backend deployment

🟩 Sprint 8 (Week 15): Web Integration & Final Deployment

Activities:

- Develop frontend interface
- Connect frontend with API
- End-to-end testing
- Deployment & final presentation

Deliverables:

- Fully functional web application
- Deployment link
- Final project presentation
- Internship completion report

📊 Final Outcome

By the end of 15 weeks, interns will have:

- Strong technical foundation
- Real-time project experience
- A complete end-to-end deployed project
- Industry-ready portfolio project

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	1 MARKS

5.3 - Project Progress Tracking:

1. Zoho Cliq Workspace Structure

🔹 Channels Setup

Create the following channels for organized communication:

📌 1. #announcements

- Official updates
- Sprint start/end notifications
- Deadlines & evaluation updates
- Meeting schedules

📌 2. #project-discussion

- Technical queries
- Implementation discussions
- Code review discussions
- Issue troubleshooting

📌 3. #Daily-updates

- Daily progress reports
- Blockers
- Completed tasks

📌 4. #resources

- Shared datasets

- Documentation links
- Recorded session links
- API documentation

📌 5. #team-specific-channels (if multiple teams)

Example:

- #team-alpha
- #team-beta

📅 2. Sprint-Based Tracking Method

Each sprint will follow a structured reporting cycle.

🟢 Daily Progress Update Format (Posted in #daily-updates)

Every intern must post:

Format:

Date:

Sprint:

Tasks Completed:

Tasks In Progress:

Blockers (if any):

Plan for Tomorrow:

🔵 Weekly Sprint Review (Every Weekend)

Mentor will post:

- Sprint Goals
- Completed Milestones
- Pending Tasks
- Risk Areas

- Next Week Targets

📋 3. Task Tracking System

Option 1: Zoho Cliq Tasks Feature

Use built-in task management in Zoho Cliq:

- Assign tasks to interns
- Set deadlines
- Track completion status
- Add task priority (High / Medium / Low)

Task Status Workflow:

- 🟡 **To Do**
- 🔵 **In Progress**
- 🟢 **Completed**
- 🔴 **Blocked**

Option 2: Zoho Projects Integration (Optional)

For advanced tracking, integrate with:

- Zoho Projects

Use:

- Kanban Board
- Gantt Chart
- Milestone tracking
- Automated reminders

📊 4. Sprint Progress Monitoring Dashboard

Track the following metrics weekly:

- % Tasks Completed

- API Development Progress
- Model Accuracy Improvement
- Deployment Readiness
- Bug Count
- Attendance in Live Sessions

Mentor shares a weekly progress summary in:
👉 **#announcements channel**

📈 5. Milestone Tracking Structure

Milestone	Week	Status	Owner	Remarks
Data Collection	Week 7–8	☐	Team	
Data Preprocessing	Week 9–10	☐	Team	
Model Building	Week 11–12	☐	Team	
API Integration	Week 13–14	☐	Team	
Web Integration	Week 15	☐	Team	

🚨 6. Escalation Process

If blocker > 24 hours:

- Post in #project-discussion
- Tag mentor
- If unresolved → Schedule quick call via Zoho Cliq
- Update resolution summary in channel

🏆 7. Performance Evaluation Criteria

Evaluation will be based on:

- Daily update consistency
- Sprint milestone completion
- Code quality
- Participation in discussions
- Final project delivery
- Timely submissions

📅 Weekly Workflow Summary

Day	Activity
Monday	Sprint planning post
Tue–Thu	Development & daily updates
Friday	Progress review
Saturday	Sprint demo
Sunday	Feedback & planning

✅ Final Outcome

Using Zoho Cliq ensures:

- Clear communication
- Accountability
- Structured sprint tracking
- Real-time problem resolution
- Professional workflow management

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

DATE	28-02-2026
TEAM ID	LTVIP2026TMID90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	1 MARKS

5.4 - Team management Tools for Agile Planning (Jira):

🧩 What is Jira?

Jira is an Agile project management and issue-tracking tool developed by Atlassian. It supports Scrum and Kanban methodologies, helping teams plan, track, and release software efficiently.

🏗 Jira Project Structure for Internship

1️⃣ Project Creation

Create a project with:

- Project Name: Internship Capstone Project
- Template: Scrum (Recommended)
- Project Type: Software Development

📌 Issue Types Configuration

Define standard issue types:

- 🟢 Epic – Major project phases
- 🔵 Story – Feature or functionality
- 🟡 Task – Smaller implementation steps
- 🔴 Bug – Errors or defects
- ⚪ Sub-task – Breakdown of tasks

📁 Suggested Epics (Based on Your Milestones)

Epic	Description
Data Collection	**Dataset gathering & validation**

Data Preprocessing	**Cleaning & feature engineering**
Model Development	**ML training & evaluation**
API Integration	**Backend & model API**
Web Integration	**Frontend & deployment**

📅 Sprint Planning Structure

🔹 Sprint Duration

- 2 Weeks per Sprint
- Total: 4–5 Sprints (Project Phase)

🔹 Sprint Workflow

1. Create Sprint in Backlog
2. Add Stories/Tasks to Sprint
3. Assign Tasks to Interns
4. Set Priority (High/Medium/Low)
5. Estimate Story Points
6. Start Sprint

🔄 Workflow Status Configuration

Customize workflow:

- 🟡 To Do
- 🔵 In Progress
- 🟣 In Review
- 🟢 Done
- 🔴 Blocked

This ensures transparent tracking of task movement.

📊 Agile Boards in Jira

1️⃣ Scrum Board

Used for:

- Sprint planning
- Daily standups
- Tracking sprint progress

2️⃣ Kanban Board (Optional)

Used for:

- Continuous workflow
- API & bug tracking

📈 Agile Reports for Monitoring

Jira provides built-in reports:

- 📊 Burndown Chart – Sprint progress tracking
- 📈 Velocity Chart – Team performance
- 📋 Sprint Report – Completed vs pending work
- 🐞 Bug Report – Defect tracking

Mentors can review reports weekly to evaluate progress.

👥 Role-Based Access

Define permissions:

Role	Responsibilities
Project Admin	Configure board & workflow
Scrum Master	Sprint planning & review
Developer (Intern)	Task implementation

Reviewer	Code & feature review

🧠 Daily Standup Format (Using Jira Board)

Each intern updates:

- What was completed yesterday
- What will be done today
- Any blockers

Tasks must be moved across workflow stages accordingly.

🔐 Integration Capabilities

Jira can integrate with:

- GitHub (code tracking)
- CI/CD tools
- Slack or Zoho Cliq (notifications)
- Confluence (documentation)

📅 Example Sprint Breakdown (2 Weeks)

Sprint Goal: Complete Data Preprocessing

Planned Stories:

- Clean missing values
- Outlier detection
- Feature scaling
- Train-test split

Expected Outcome:
Clean dataset ready for model training.

✅ Benefits of Using Jira

- Structured Agile Planning
- Clear Accountability

- Transparent Sprint Tracking
- Performance Analytics
- Real-time Status Visibility
- Industry-standard project management tool

📌 Final Outcome

By implementing Jira for Agile planning, the internship project will follow:

- Scrum-based sprint execution
- Organized backlog management
- Efficient task delegation
- Measurable progress tracking
- Professional software development workflow

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

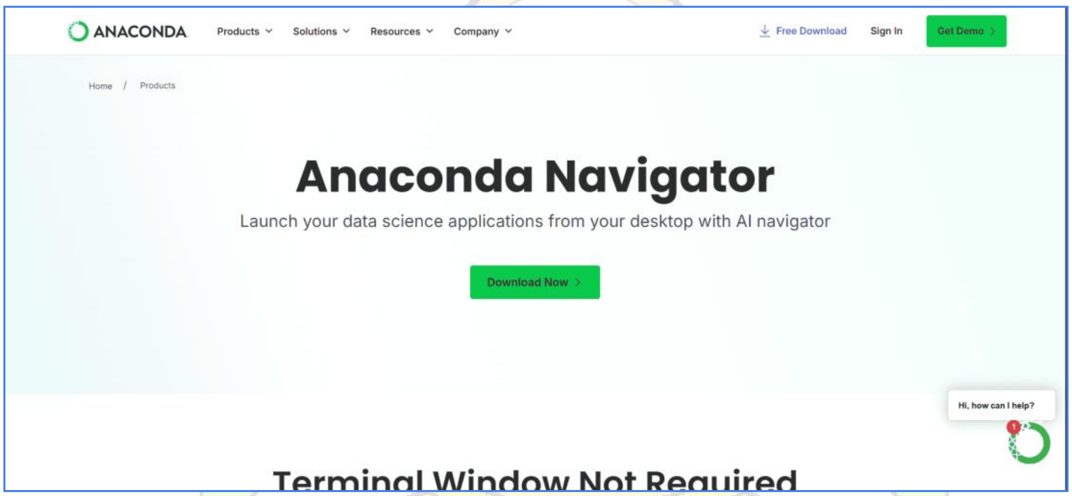
DATE	28-02-2026
TEAM ID	LTVIP2026TMIDS90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	2 MARKS

Chapter 6
Project Development

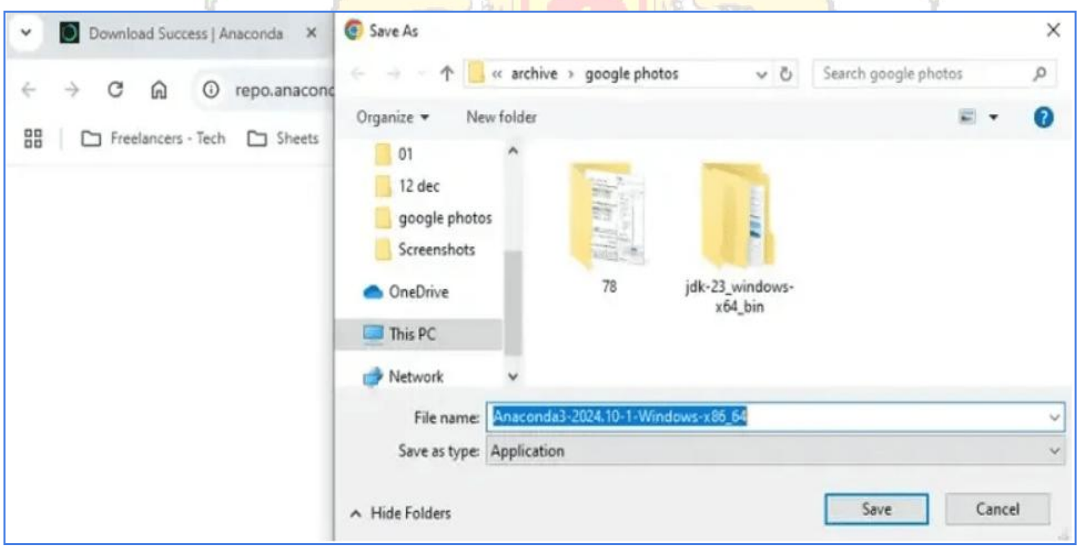
6.1 - Pre-Requisites:

→ How to install Anaconda Navigator

Step - 1: Open this link https://www.anaconda.com/products/navigator



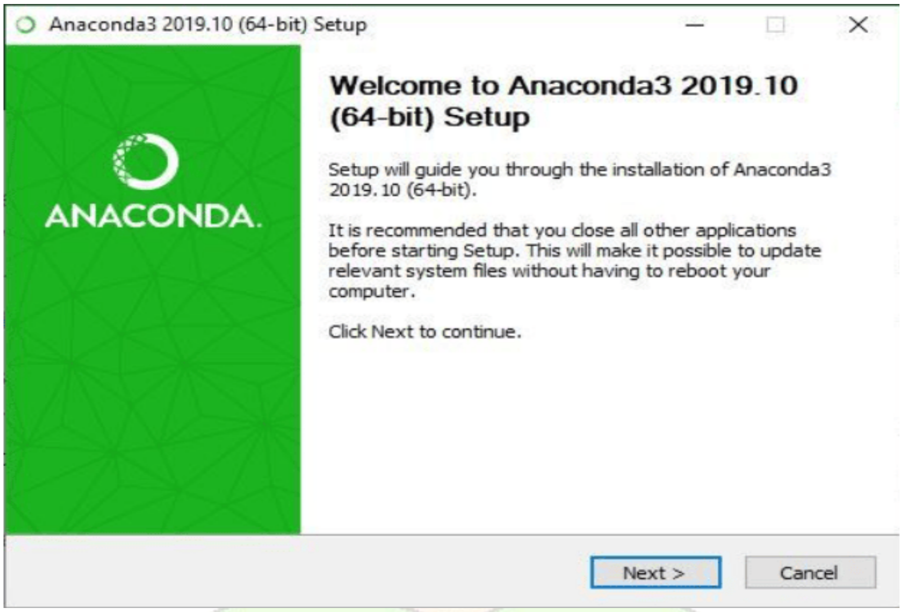
Step - 2: Click on Download button



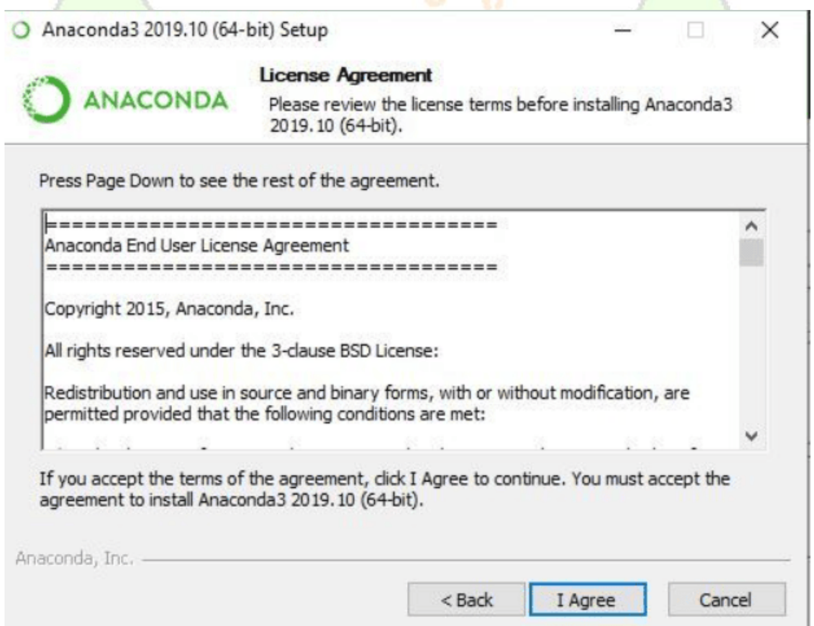
Step - 3: After downloading the Anaconda Navigator double click on the file

Step - 4: Select the Windows Installer

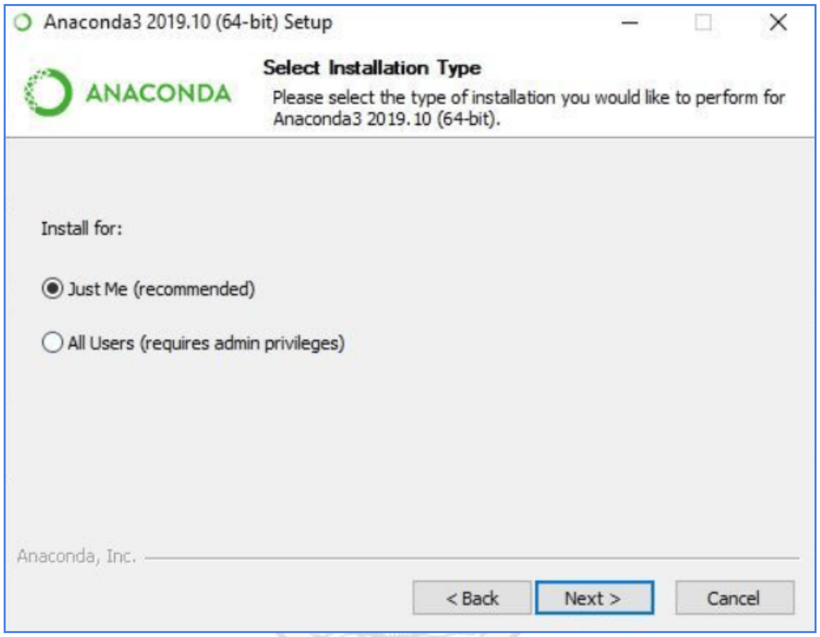
Step - 5: Begin the installation process



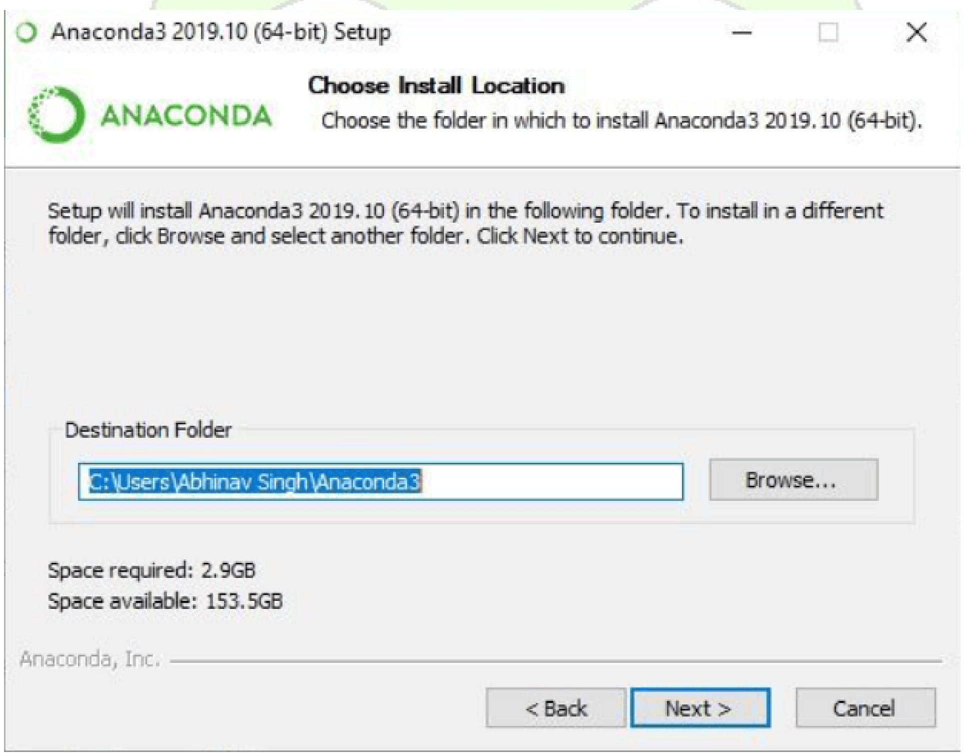
Step - 6: Getting through licence agreement



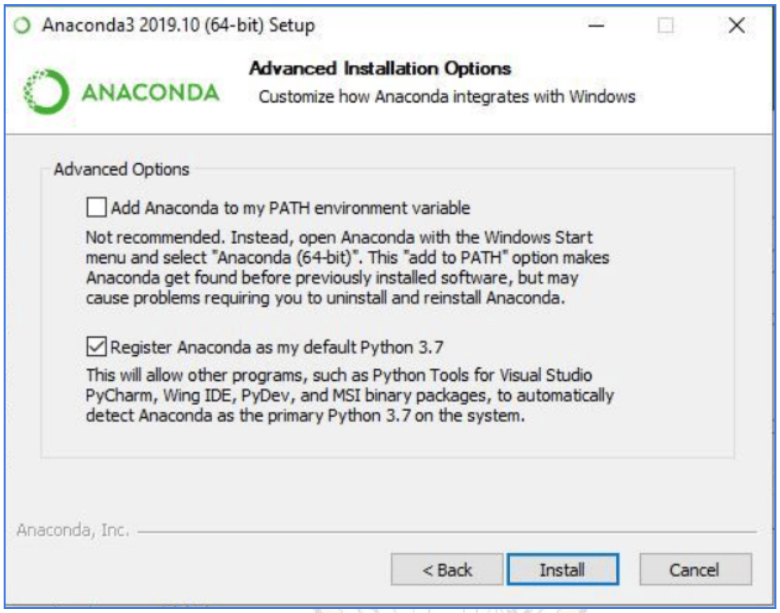
Step - 7: Select Installation Type



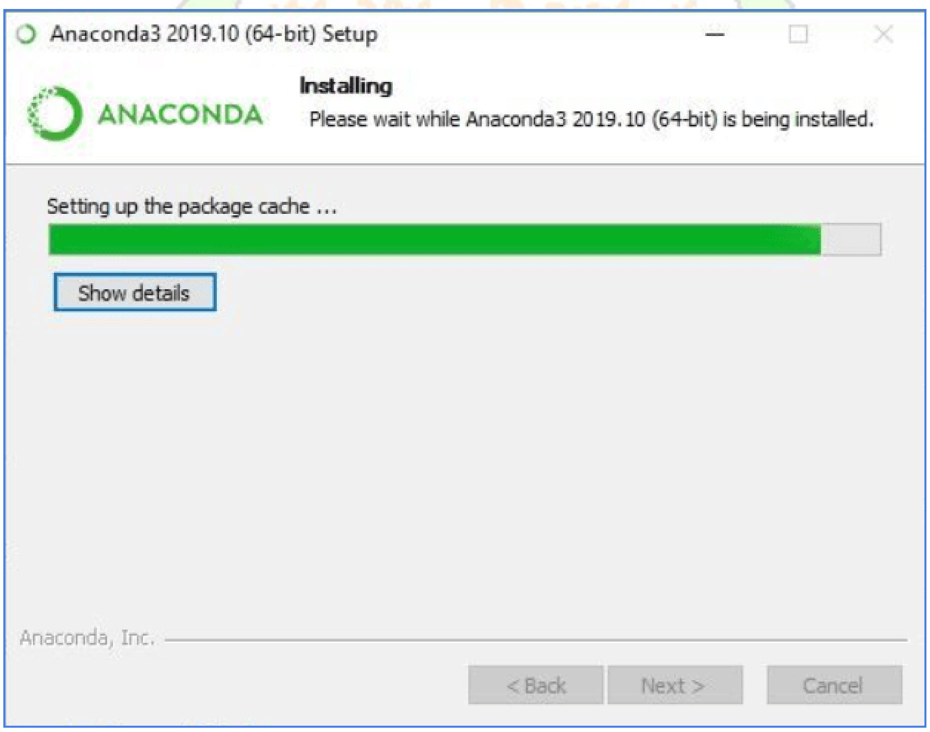
Step - 8: Select the path where you wish to install the file extractor and click "Next" to proceed ahead.



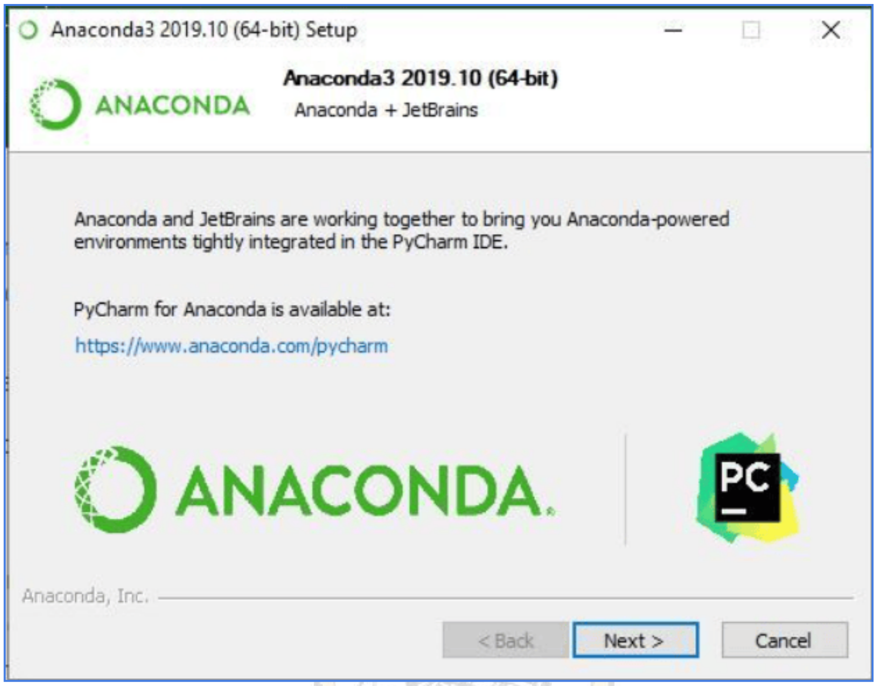
Step - 9: Advance installation options



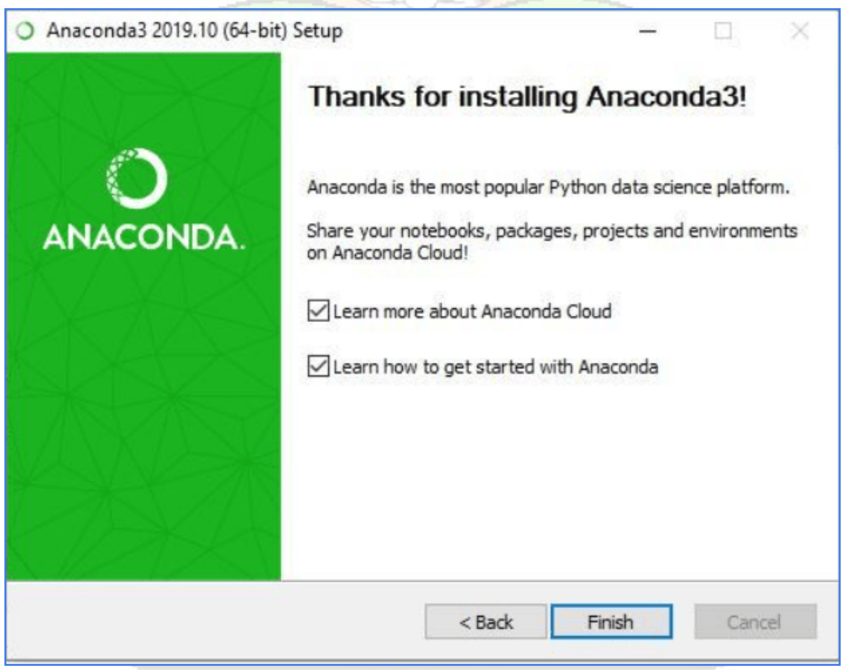
Step - 10: Click Install to start the Anaconda Installation process.



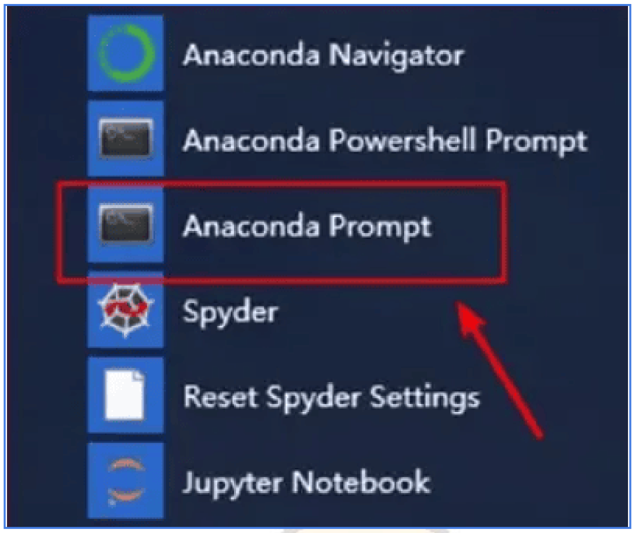
Step - 11: Recommendation to install PyCharm



Step - 12: Once the installation gets complete, click Finish to complete the process.



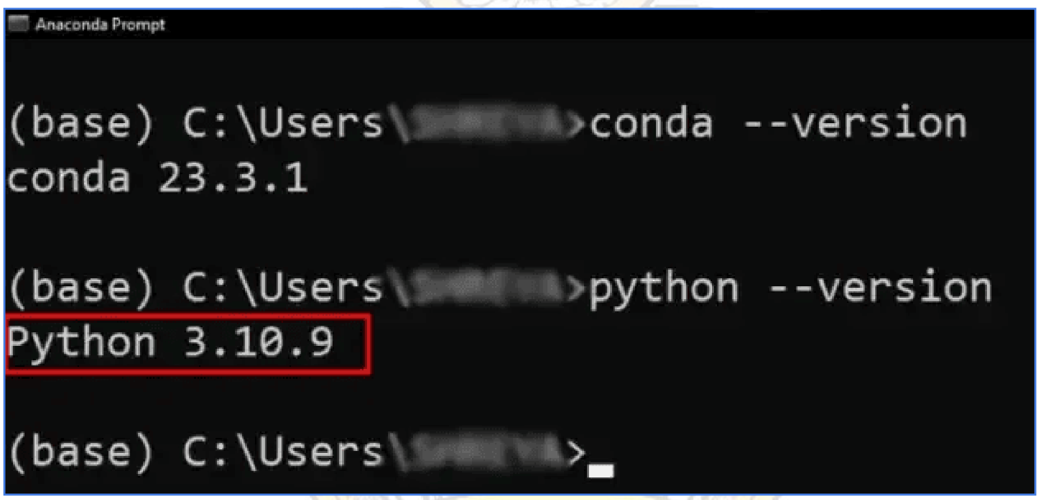
Step - 13: Click on the Start Menu and search for "Anaconda Prompt" and click to open it.



Step - 14: Run the Program to check for the Anaconda Version

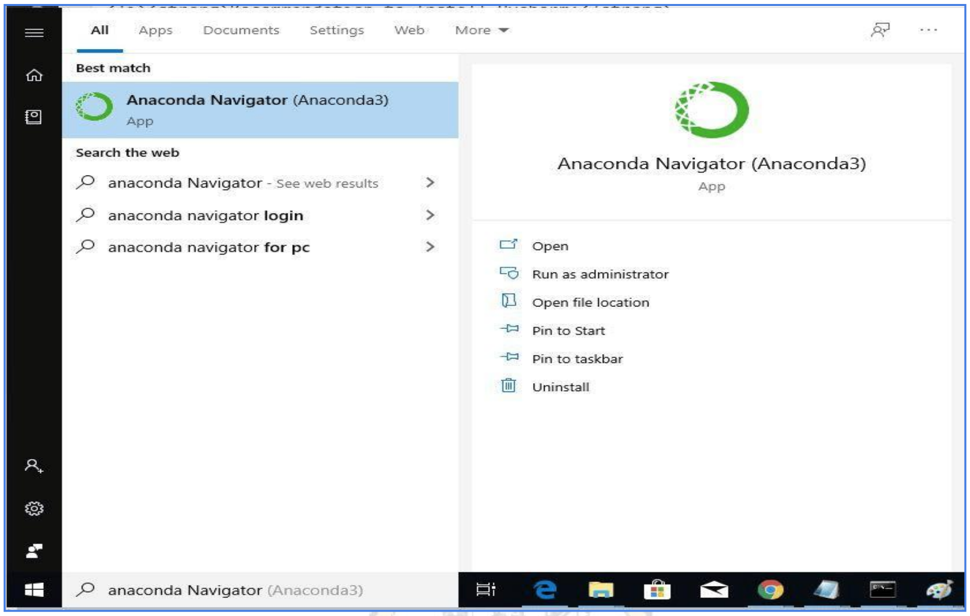
Type the following command to check for the installed version of conda:

Conda --version

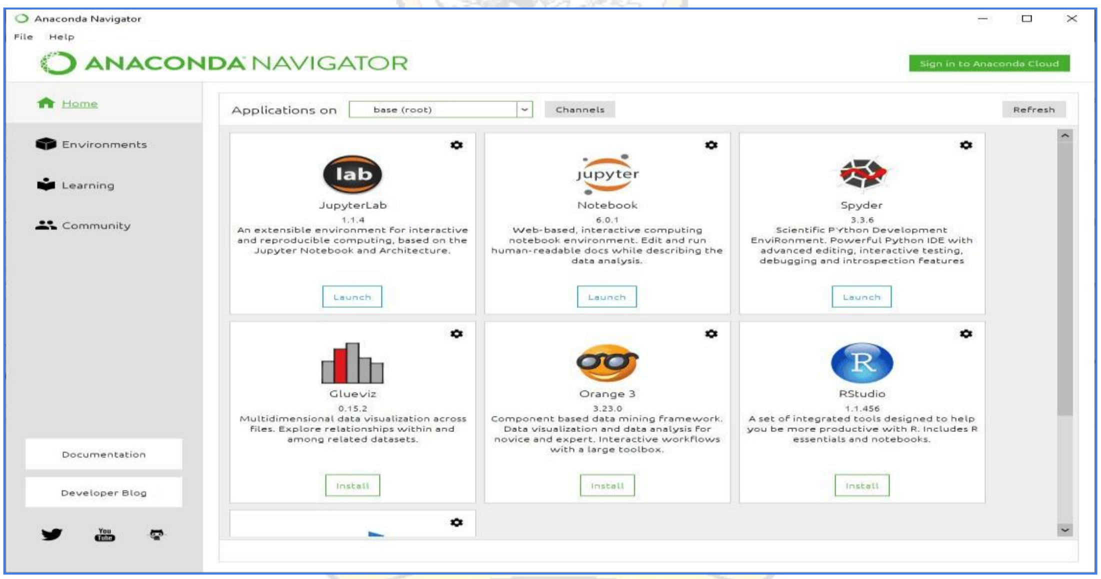


Step - 15: Access Anaconda Navigator

Once the installation process is done, Anaconda can be used to perform multiple operations. To begin using Anaconda, search for Anaconda Navigator from the Start Menu in Windows PC.



Step - 16: Explore Navigator & Features

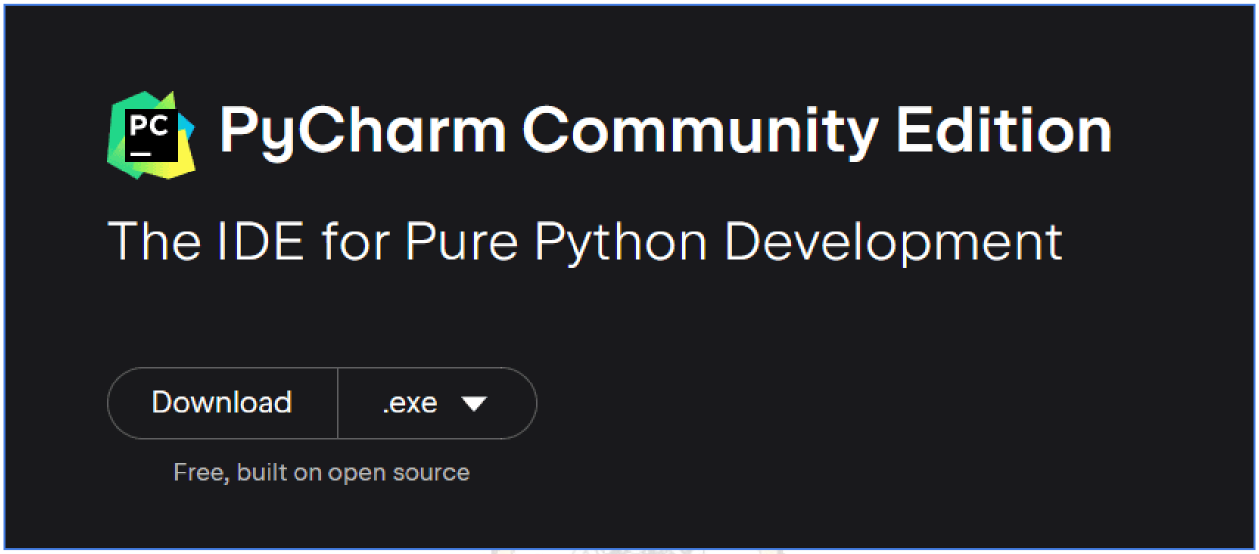
→ 

How to Install PyCharm:

Step 1: Download PyCharm

- Go to the JetBrains PyCharm download page

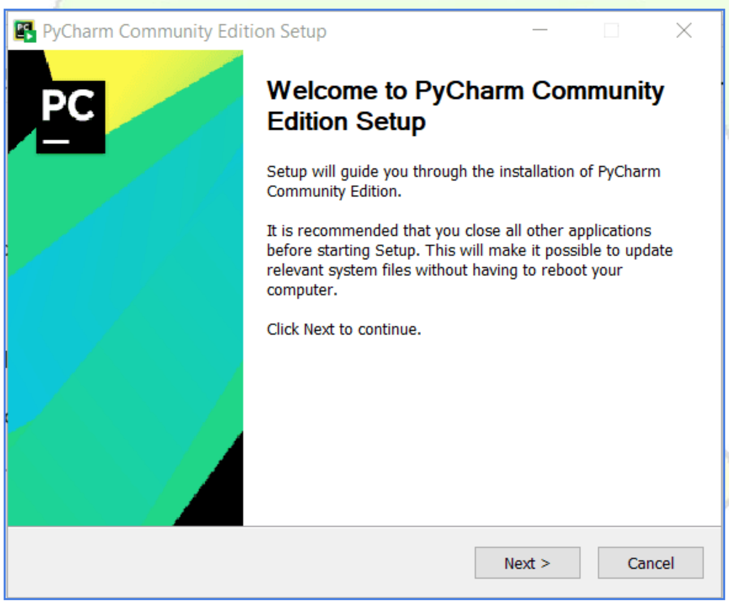
- Choose either the Community edition (free) or the Professional edition (paid).



- Download the installer according to your preference.

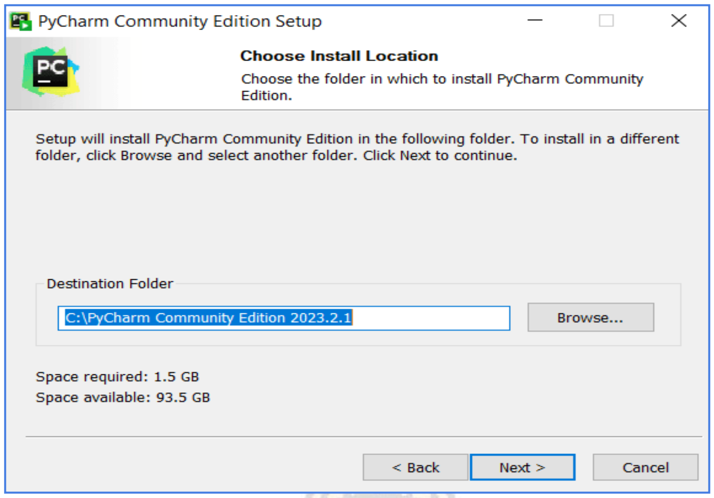
Step - 2: Run the Installer

- Locate the downloaded installer file and double-click it to run.
- Follow the on-screen instructions.



Step - 3: Choose Installation Location

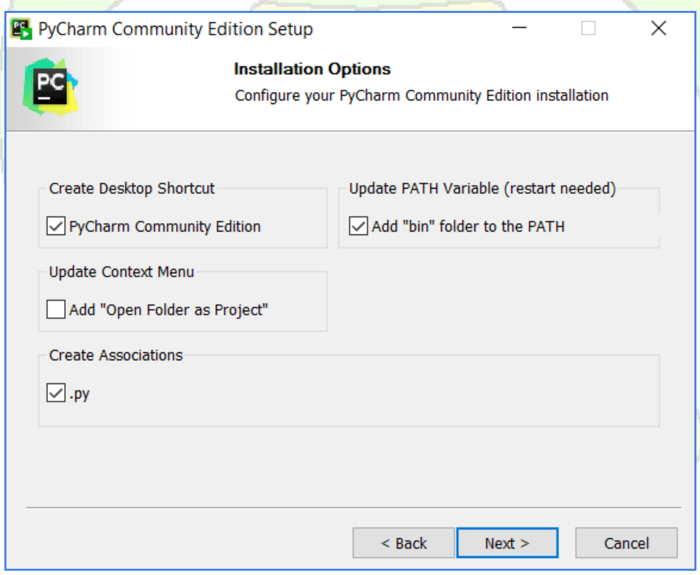
- Select the folder where you want to install PyCharm.



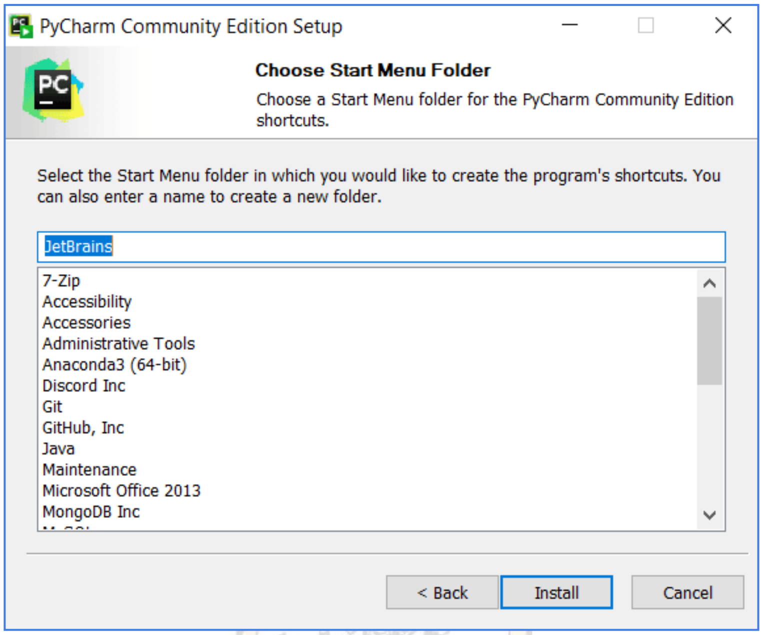
Step - 4: Select Installation Options

You can select the following options according to your preference:

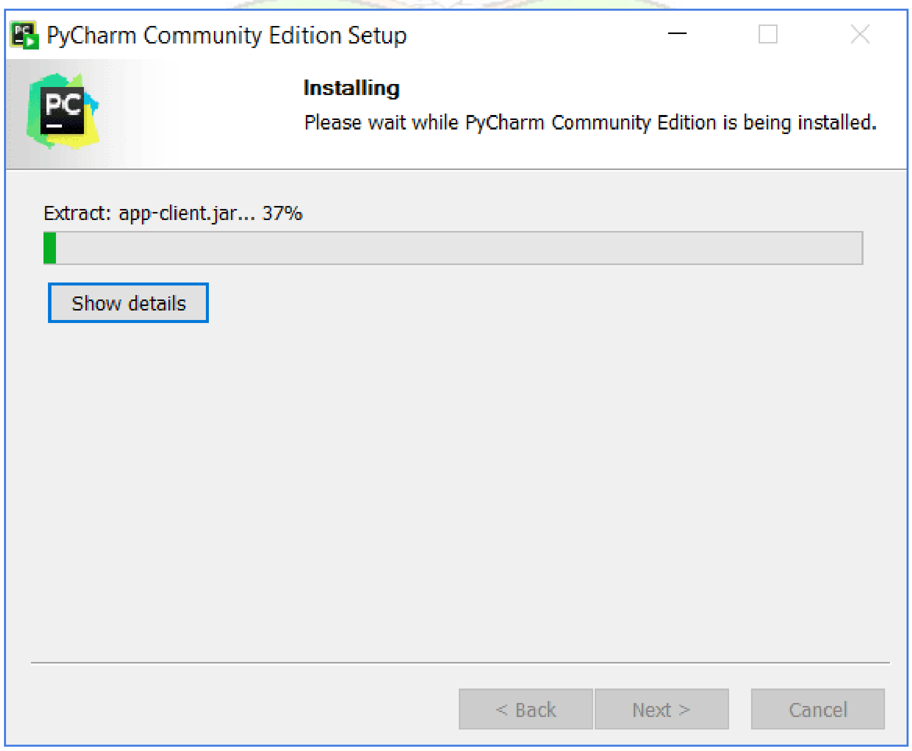
- Create Desktop Shortcut - adds a shortcut on your desktop.
- Add PyCharm to PATH - allows you to run PyCharm from the command line.
- Associate.py files with PyCharm - makes PyCharm the default editor for Python files.



- Choose folder name where your project will be saved as default.



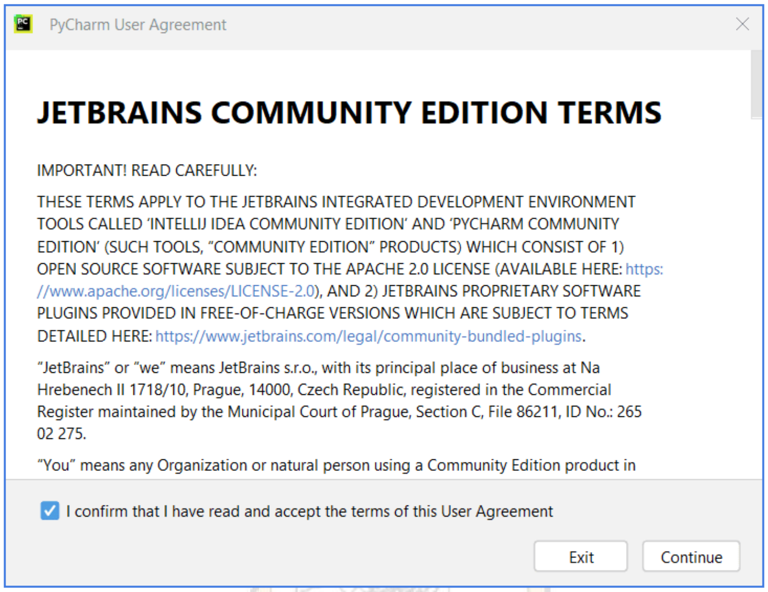
- Now it will download the PyCharm.



Step 5: Complete Installation

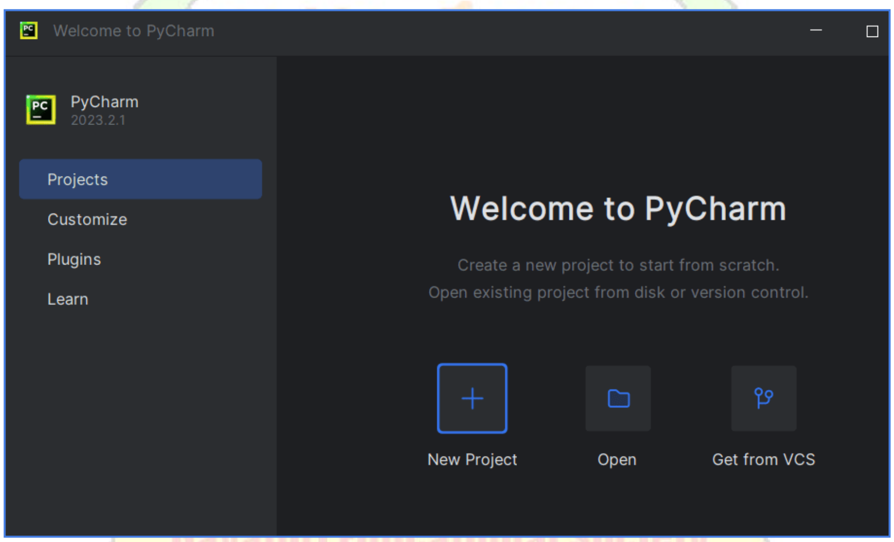
- Accept the license agreement and continue.

- PyCharm will be installed.



Step 6: Launch PyCharm

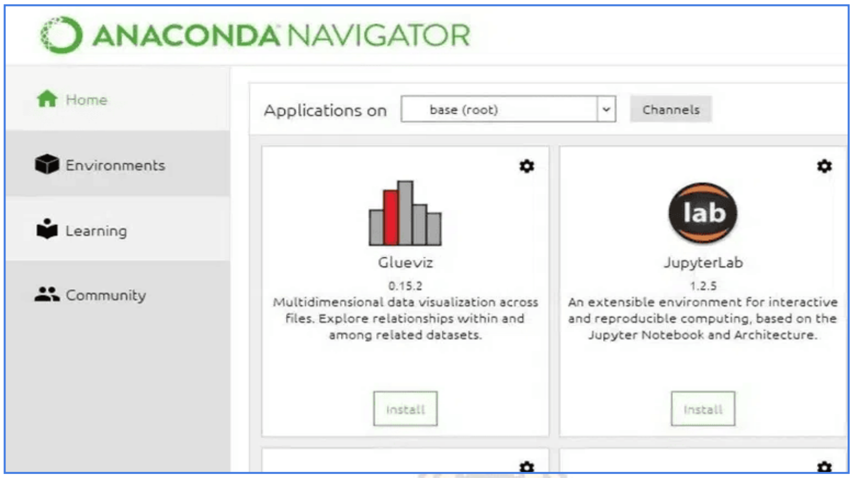
- Once installed, launch PyCharm either from the Start menu or the desktop shortcut.



→ How to open Jupyter Notebook:

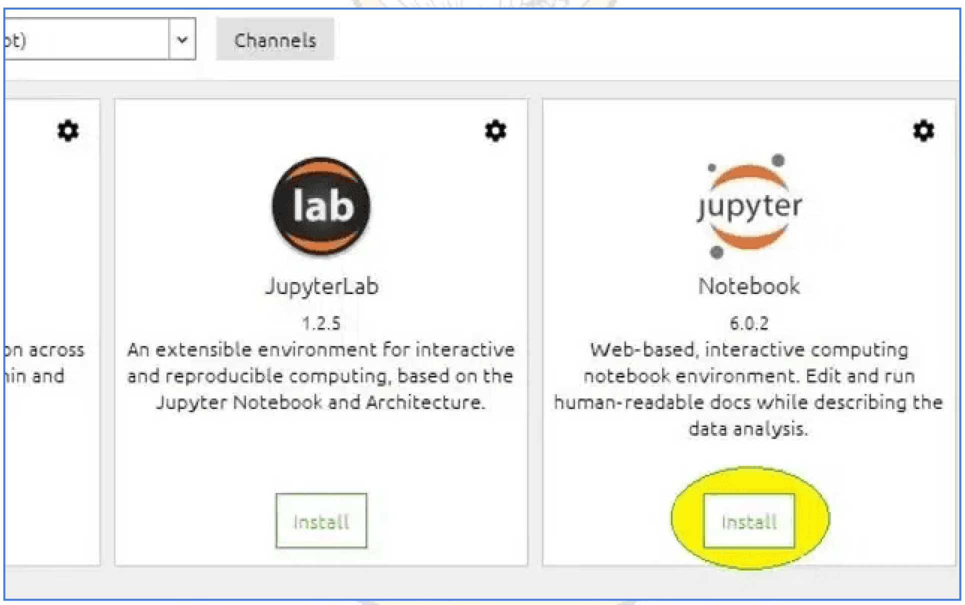
Step 1: Go to Anaconda Navigator

Firstly, Launch anaconda and click on the Install Jupyter Notebook Button.



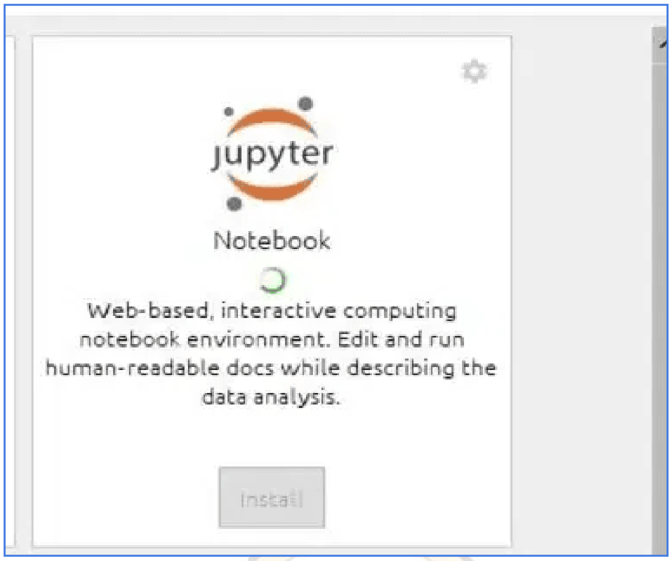
Step 2: Install Jupyter Notebook

Search for Jupyter Notebook and click on the Install button to begin with the installation process.



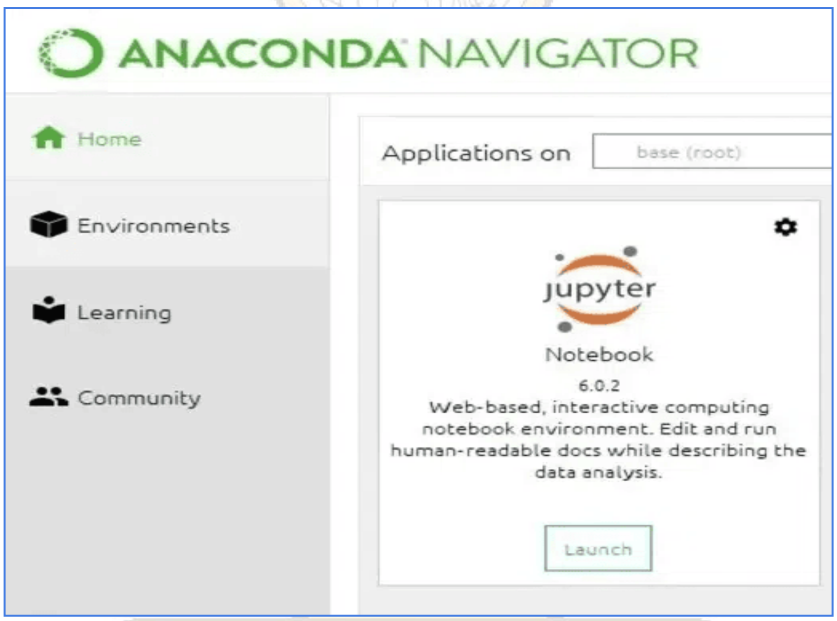
Step 3: Load Packages

Once the installation is complete, it will start loading the packages that comes along with it and click to finish the Installation.



Step 4: Launch Jupyter Notebook

Now, click on Launch button to start the Jupyter Notebook.



How to install Packages

If you are using Pycharm IDE, you can install the packages through the command prompt and follow the same syntax as above.

Type “pip install numpy” and click enter.

```
(.venv) PS D:\LT Python Project> pip install numpy
Requirement already satisfied: numpy in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (2.2.3)

[notice] A new release of pip is available: 25.1.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS D:\LT Python Project>
```

Type “pip install pandas” and click enter.

```
(.venv) PS D:\LT Python Project> pip install pandas
Requirement already satisfied: pandas in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.1)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

[notice] A new release of pip is available: 25.1.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS D:\LT Python Project>
```

Type “pip install matplotlib” and click enter.

```
(.venv) PS D:\LT Python Project> pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (2.2.3)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)

[notice] A new release of pip is available: 25.1.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS D:\LT Python Project>
```

Type “pip install scikit-learn” and click enter

```
(.venv) PS D:\LT Python Project> pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (1.7.2)
Requirement already satisfied: numpy>=1.22.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from scikit-learn) (2.2.3)
Requirement already satisfied: scipy>=1.8.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from scikit-learn) (1.16.2)
Requirement already satisfied: joblib>=1.2.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from scikit-learn) (3.6.0)

[notice] A new release of pip is available: 25.1.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS D:\LT Python Project>
```

Type “pip install Flask” and click enter.

```
(.venv) PS D:\LT Python Project> pip install Flask
Requirement already satisfied: Flask in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (3.1.2)
Requirement already satisfied: blinker>=1.9.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (3.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from Flask) (3.1.3)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from click>=8.1.3->Flask) (0.4.6)

[notice] A new release of pip is available: 25.1.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS D:\LT Python Project>
```

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

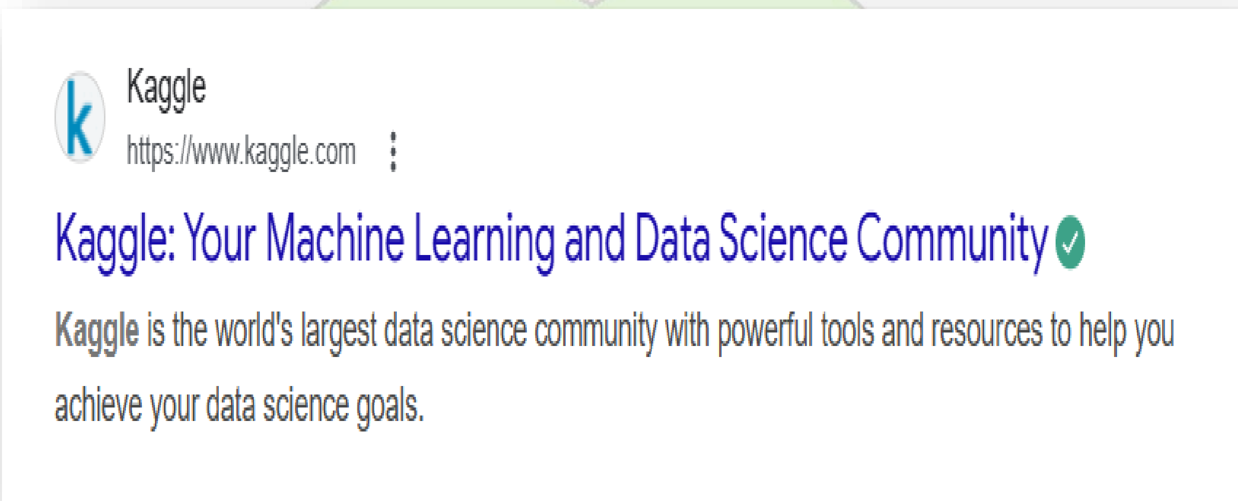
DATE	28-02-2026
TEAM ID	LTVIP2026TMIDS90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	2 MARKS

6.2 - Data Collection:

ML depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

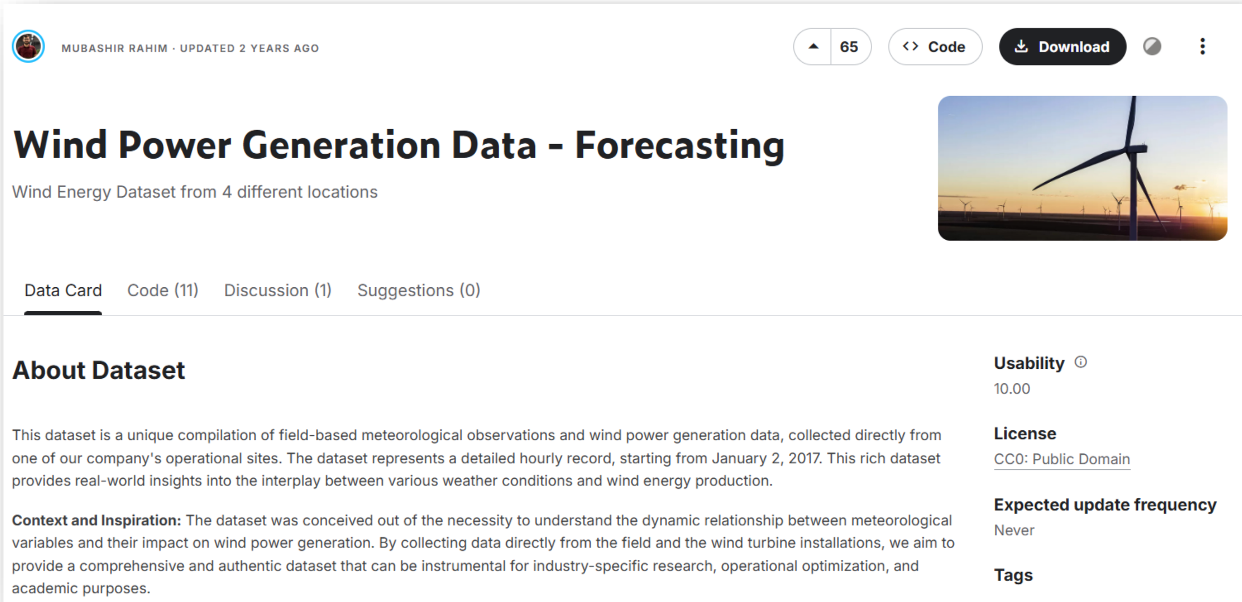
- **Download dataset /create dataset:**
  The dataset for wind energy prediction is to be collected. The dataset which is considered here will have the environmental conditions. You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository etc.

Step 1: Open Kaggle.com in Google.



Step 2: Click on the link and select Sign in with google.

Step 3: In the search bar "**Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management"** to see the dataset.

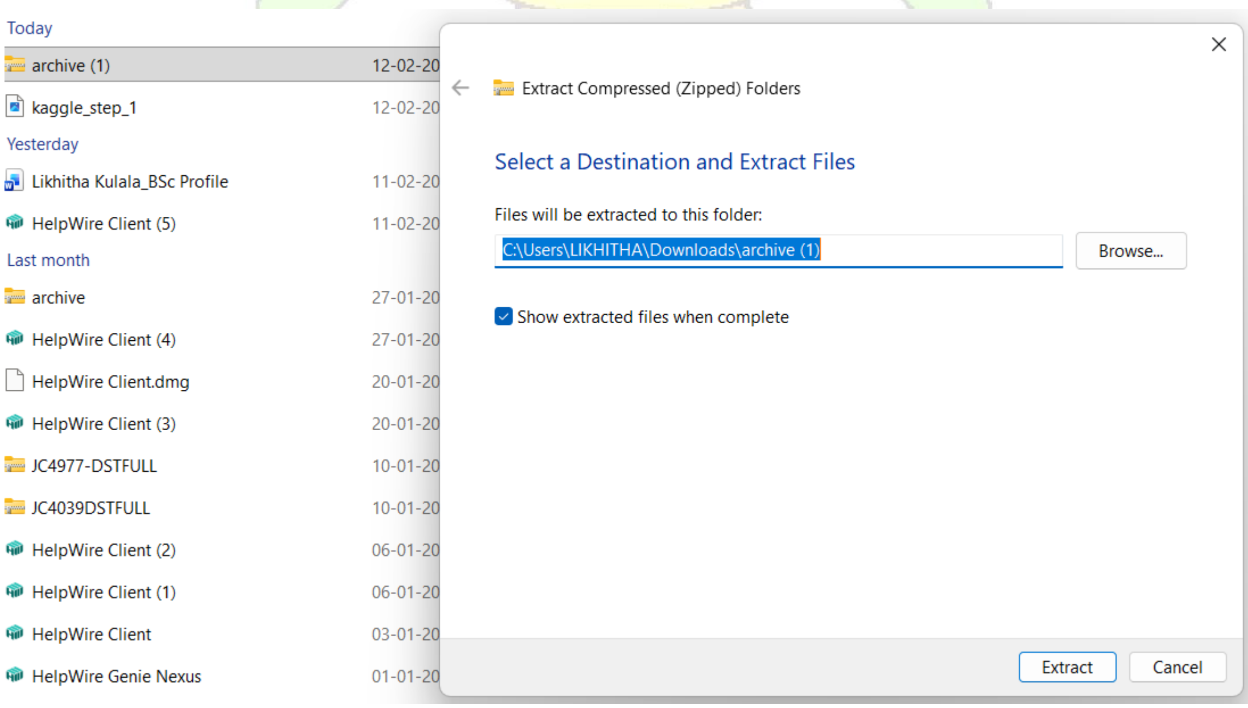


Step 4: Click the download option to download the dataset.

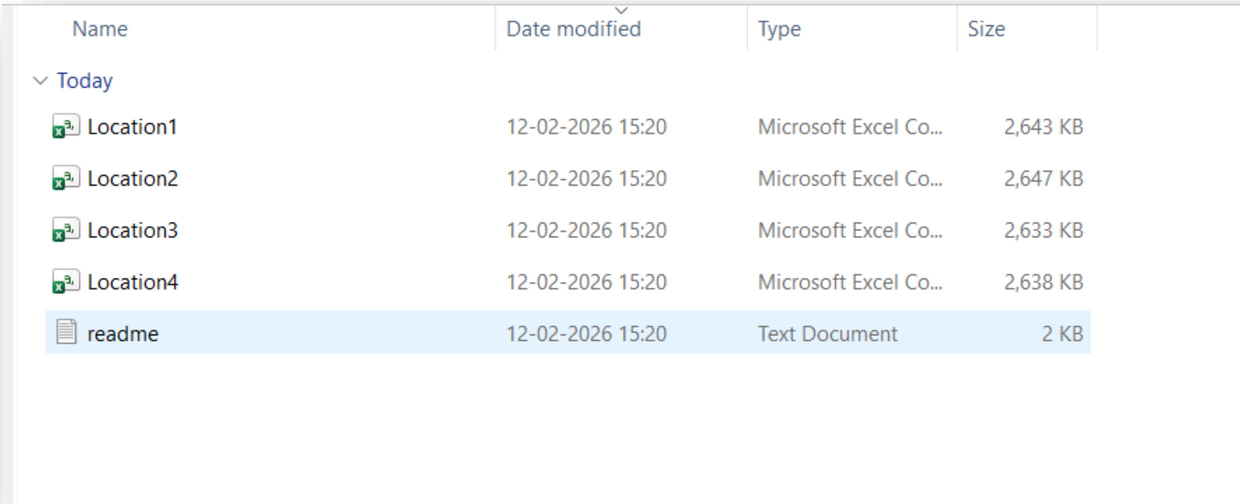
Step 5: Click on Download dataset as zip (3MB).

Step 6: Open the downloaded file.

Step 7: Unzip the file.



Step 8: Use the XL sheet which you have used in the project.



Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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	3 MARKS

6.3 - Data Pre-Processing:

In this milestone, we will be preprocessing the dataset that is collected. Preprocessing includes:

1. Processing the dataset.
2. Handling the null values.
3. Handling the categorical values if any.
4. Normalize the data if required.
5. Identify the dependent and independent variables.
6. Split the dataset into train and test sets.

Import required libraries:

```
# Importing Necessary Libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
```

Analyse the datasets:

Step 1: The datasets are imported as data frames using the panda's library. Rename the columns with suitable column names for better understanding.

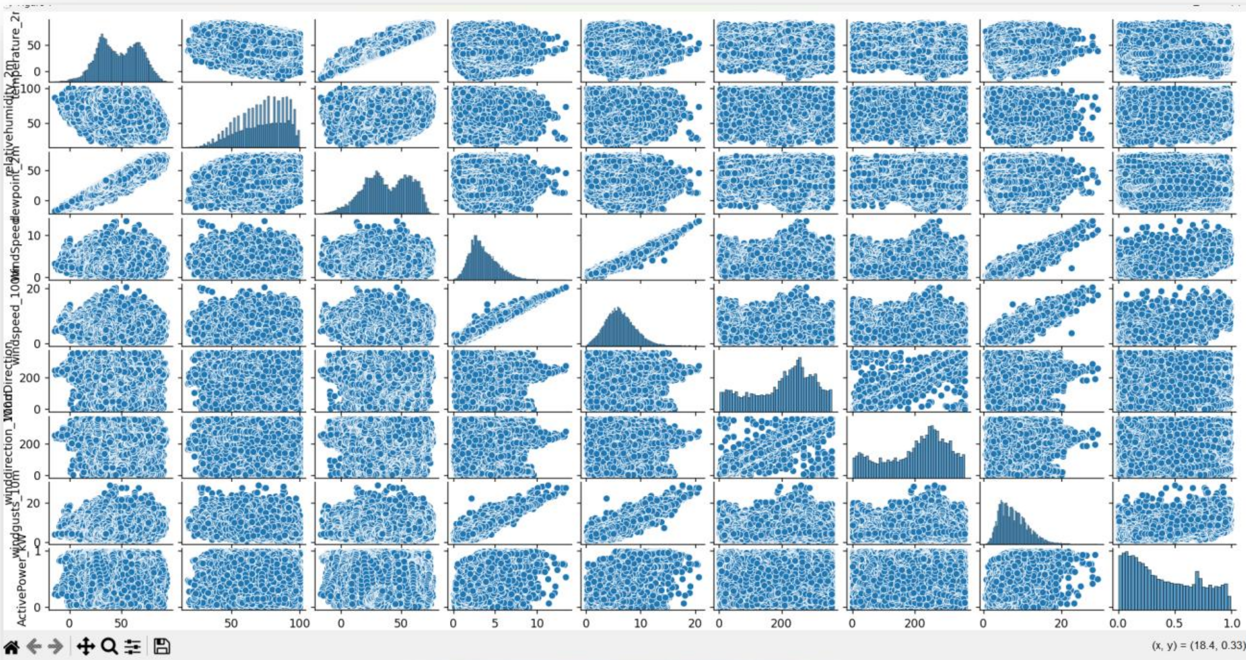
*Dataset contains the wind speed and wind direction along with the power generated.

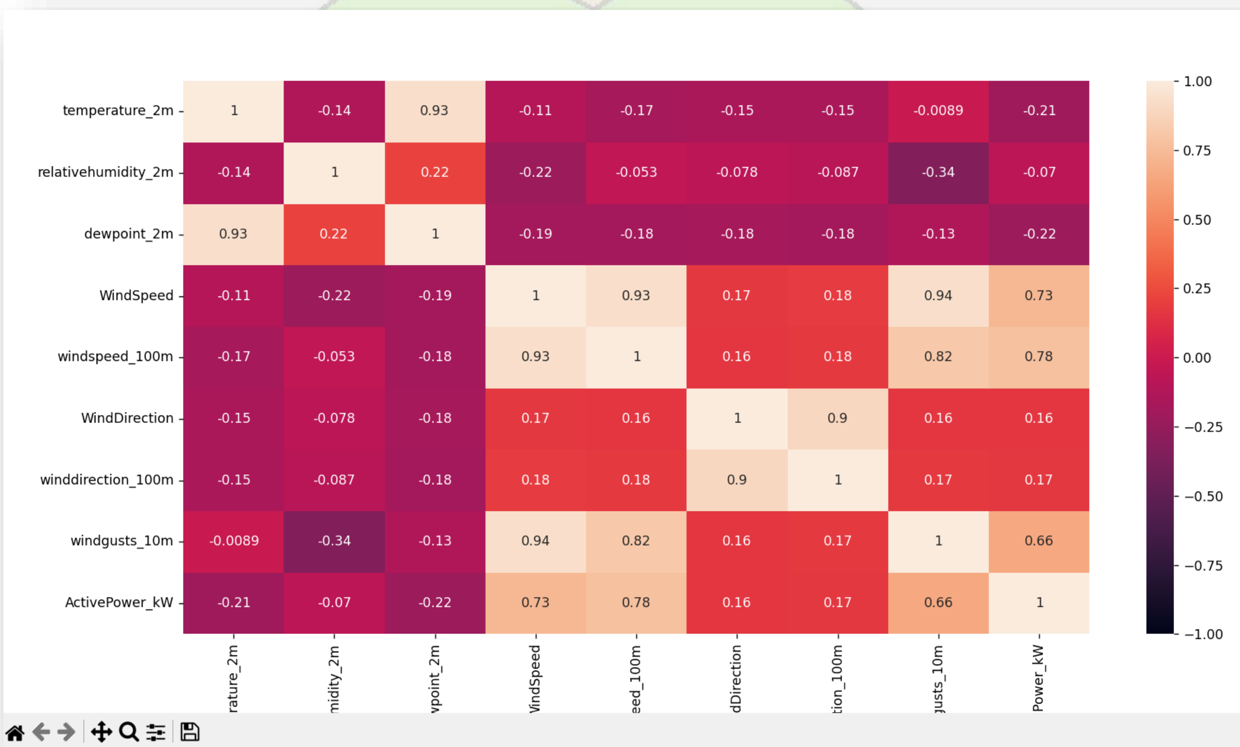
```
path = "data/Location1.csv"

df = pd.read_csv(path, encoding="latin1")

df.columns = df.columns.str.strip()

df.rename(columns={

    'Date/Time': 'Time',
```

```
            'LV ActivePower (kW)': 'ActivePower_kW',
            'Wind Speed (m/s)': 'WindSpeed',
            'Wind Direction (°)': 'WindDirection'
        }, inplace=True)
```

Step 2: Check the correlation between the columns for dimensionality reduction (knowing which columns are necessary and which are not)

```
sns.pairplot(df)
plt.figure(figsize=(10, 8))
corr = df.corr(numeric_only=True)
ax = sns.heatmap(corr, vmin=-1, vmax=1, annot=True)
bottom, top = ax.get_ylim()
plt.show();
ax.set_ylim(bottom + 0.5, top - 0.5)
print(corr)
df["Time"] = pd.to_datetime(df["Time"], errors="coerce")
```




temperature_2m
relativehumidity_2m
dewpoint_2m
WindSpeed
windspeed_100m
WindDirection
winddirection_100m
windgusts_10m
ActivePower_kW
1 -0.14 0.93 -0.11 -0.17 -0.15 -0.15 -0.0089 -0.21
-0.14 1 0.22 -0.22 -0.053 -0.078 -0.087 -0.34 -0.07
0.93 0.22 1 -0.19 -0.18 -0.18 -0.18 -0.13 -0.22
-0.11 -0.22 -0.19 1 0.93 0.17 0.18 0.94 0.73
-0.17 -0.053 -0.18 0.93 1 0.16 0.18 0.82 0.78
-0.15 -0.078 -0.18 0.17 0.16 1 0.9 0.16 0.16
-0.15 -0.087 -0.18 0.18 0.18 0.9 1 0.17 0.17
-0.0089 -0.34 -0.13 0.94 0.82 0.16 0.17 1 0.66
-0.21 -0.07 -0.22 0.73 0.78 0.16 0.17 0.66 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Splitting data into independent and dependent variables

In this activity, the dependent and independent variables are to be identified. The independent columns are considered as x and the dependent column as y.

After identifying the dependent and independent variables, the dataset now has to be split into two sets, one set is used for training the model and the second set is used for testing how good the model is built. The split ratio we consider is 80% for training and 20% for testing.

```
df["Hour"] = df["Time"].dt.hour
df["Month"] = df["Time"].dt.month
df["DayOfYear"] = df["Time"].dt.dayofyear
required_cols = ["ActivePower_kW", "WindSpeed"]
if "Theoretical_Power_Curve (KWh)" in df.columns:
    required_cols.append("Theoretical_Power_Curve (KWh)")
    print("✓ Using Theoretical Power Curve feature")
else:
    print("⚠ Theoretical Power Curve column NOT found")
df = df.dropna(subset=required_cols)
features = ["WindSpeed", "Hour", "Month", "DayOfYear"]
if "Theoretical_Power_Curve (KWh)" in df.columns:
    features.insert(1, "Theoretical_Power_Curve (KWh)")
X = df[features]
y = df["ActivePower_kW"]
train_X, val_X, train_y, val_y = train_test_split(
    X, y, test_size=0.25, shuffle=False
)
```

[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

6.4 - Model Building:

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms can be chosen according to the objective. As the dataset which we are using is a Regression dataset so you can use the following algorithms

- Linear Regression
- Random Forest Regression / Classification
- Decision Tree Regression / Classification

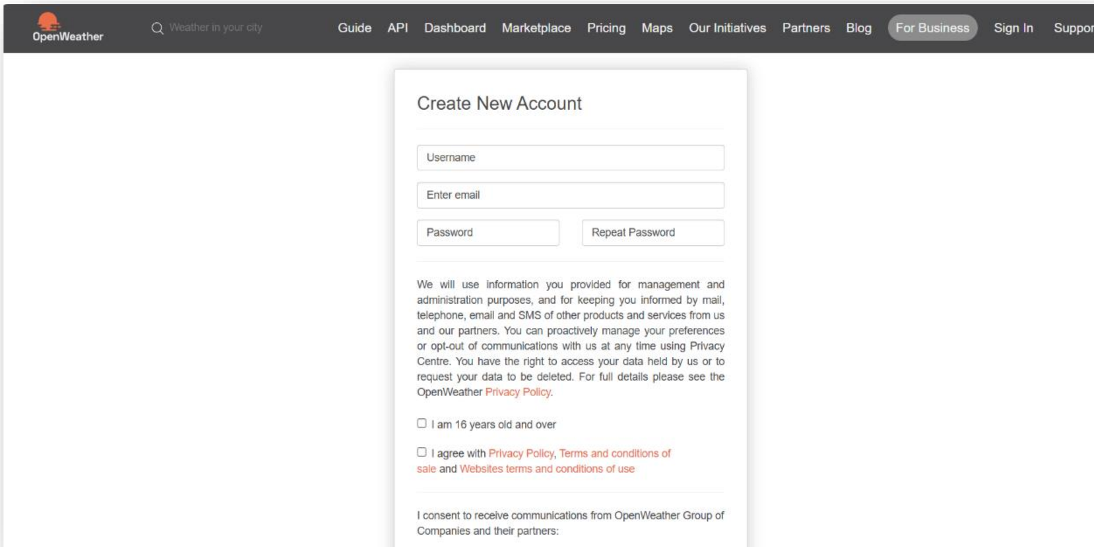
Choose the appropriate model:

```
train_X, val_X, train_y, val_y = train_test_split(
X, y, test_size=0.25, shuffle=False)
model = RandomForestRegressor(
n_estimators=300,
max_depth=6,
min_samples_leaf=5,
random_state=42,
n_jobs=-1
)
model.fit(train_X, train_y)
preds = model.predict(val_X)
print("MAE:", mean_absolute_error(val_y, preds))
print("R² :", r2_score(val_y, preds))
```

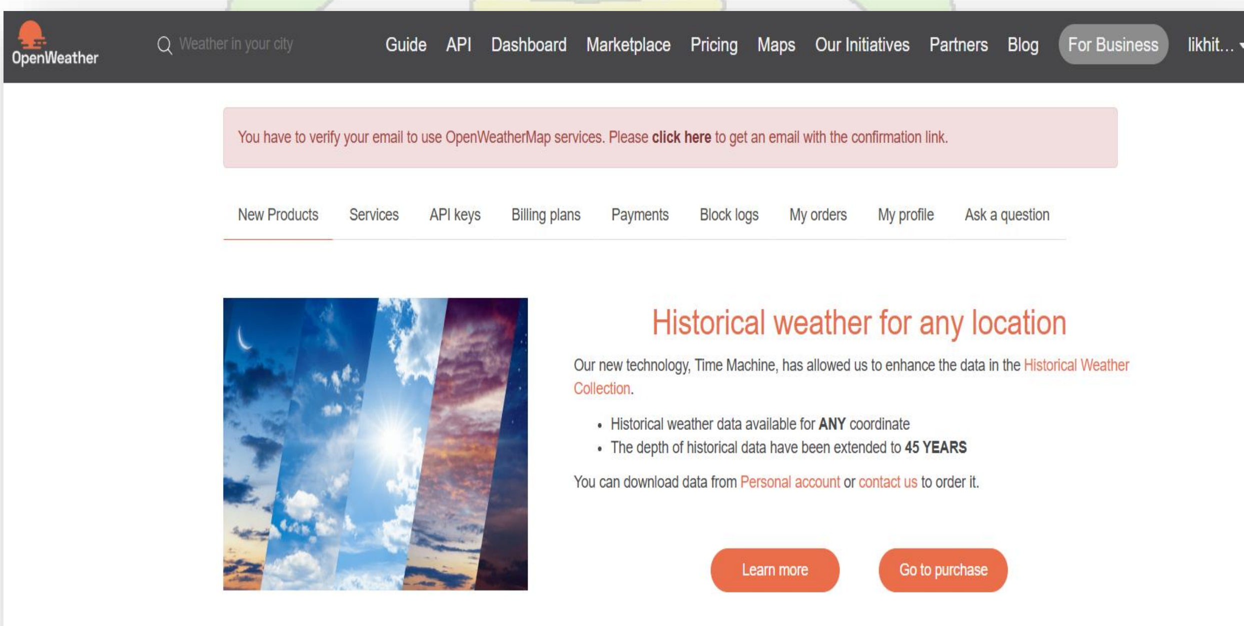
API Integration

Step 1: Signup for Open Weather API for current weather forecasting. To signup

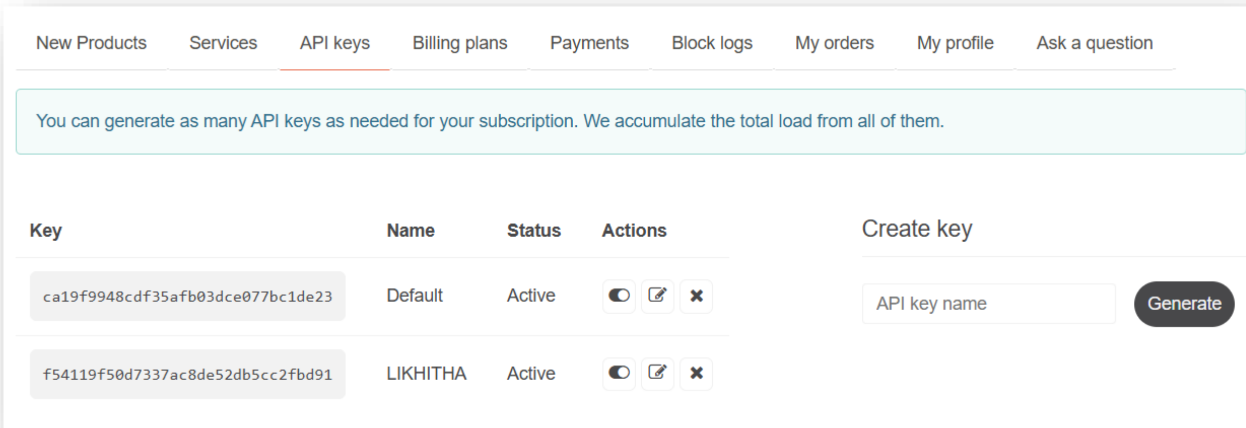
Link: - [*https://home.openweathermap.org/users/sign_up*](https://home.openweathermap.org/users/sign_up)



Step 2: After verification and subscription within 24 hours the API key will be activated.



Step 3: The API Key can be used to get the weather forecast of any of the cities known. The city is passed with parameter q and apikey is to be given with the parameter appid. An example for London city is shown below.



Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

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)
```

Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management

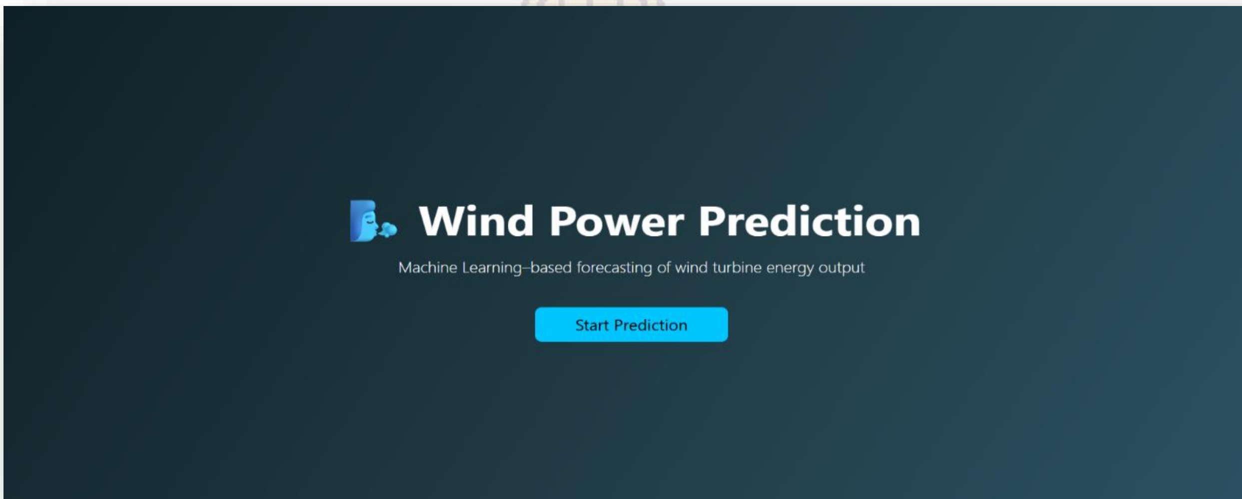
DATE	28-02-2026
TEAM ID	LTVIP2026TMIDS90651
PROJECT NAME	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
MAXIMUM MARKS	5 MARKS

Chapter 7

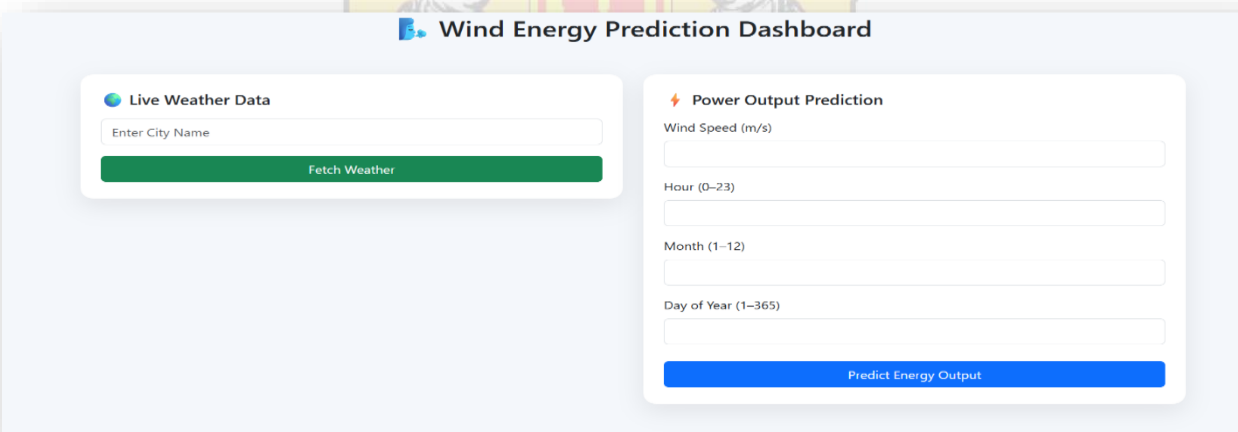
Functional and Performance Testing

7.1 - Functional and Performance Testing:

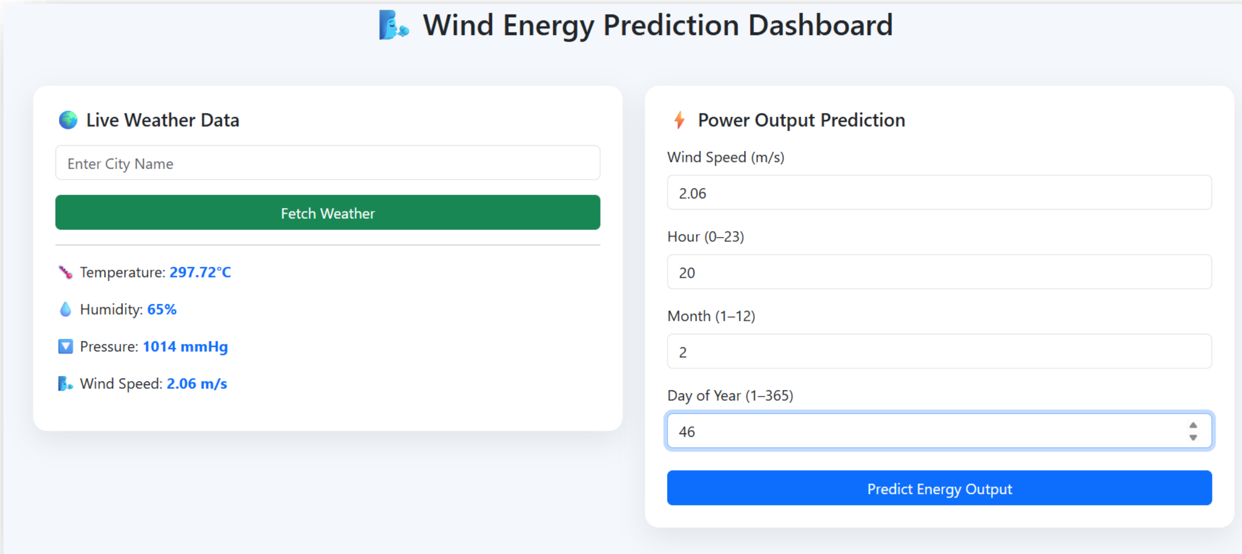
Home Screen:

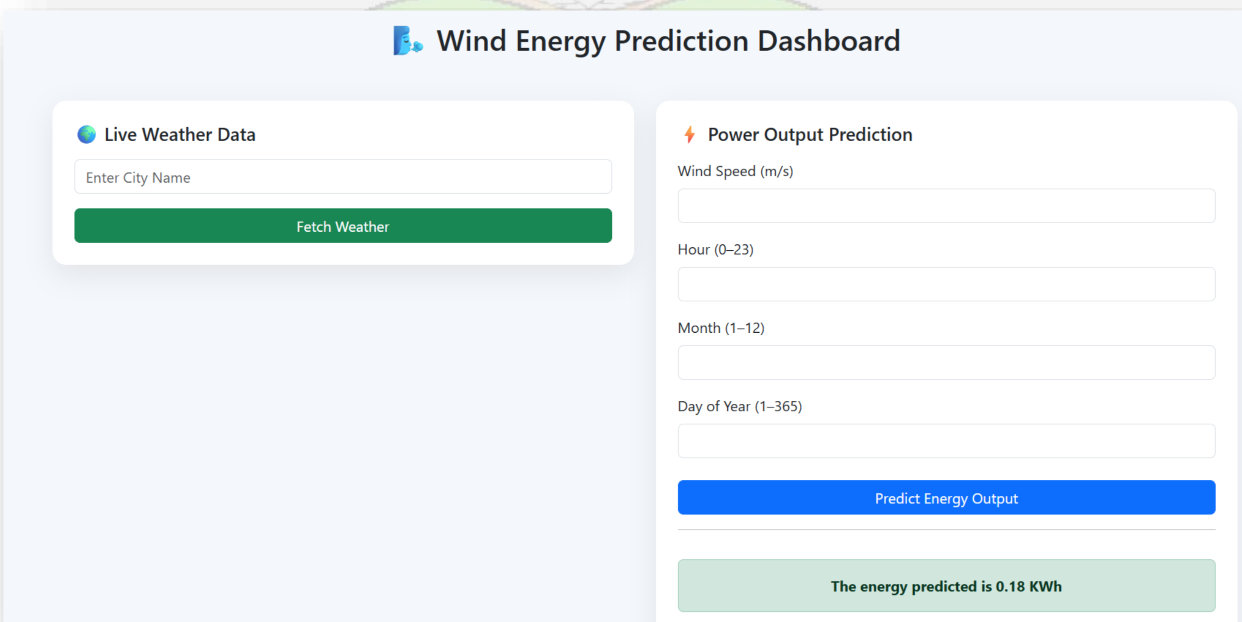


Prediction Screen:



Output Screen:





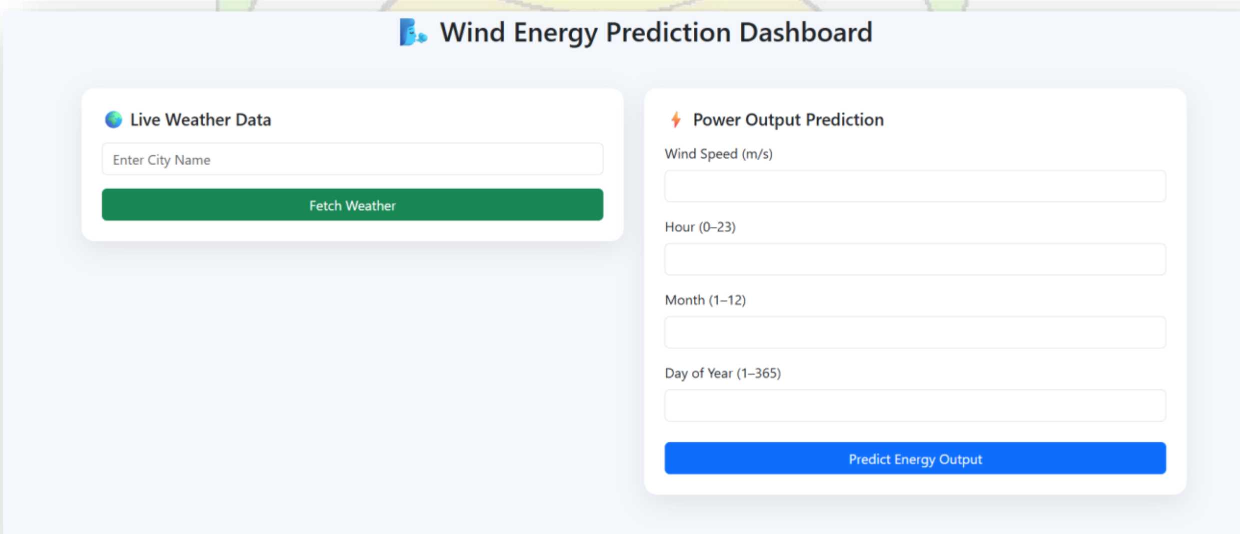
Chapter 8

Results

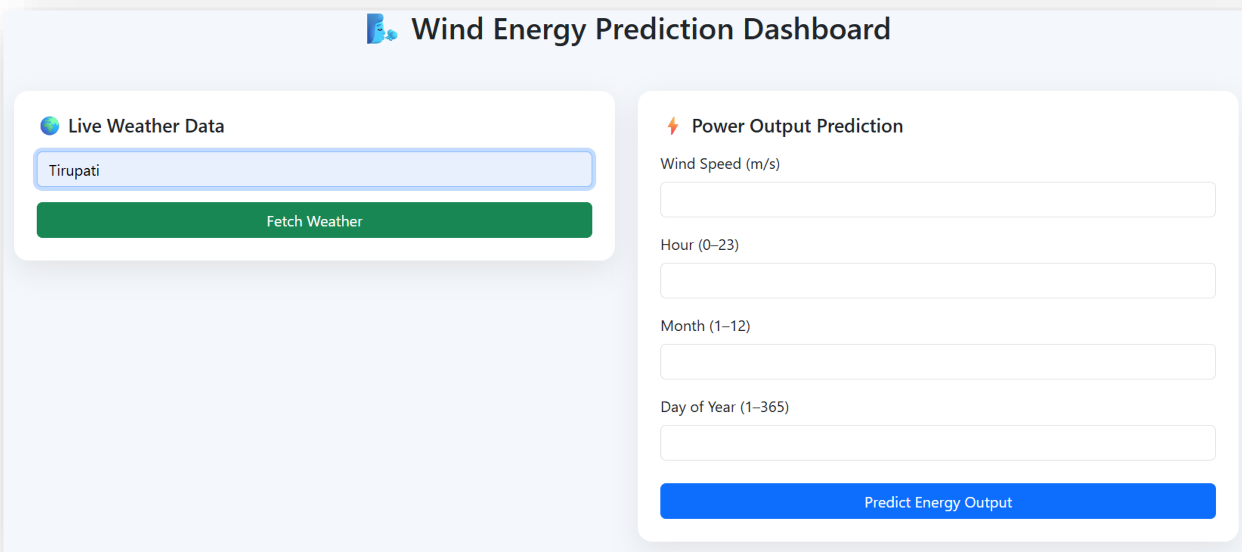
8.1 - Output Screens:

Home Page:

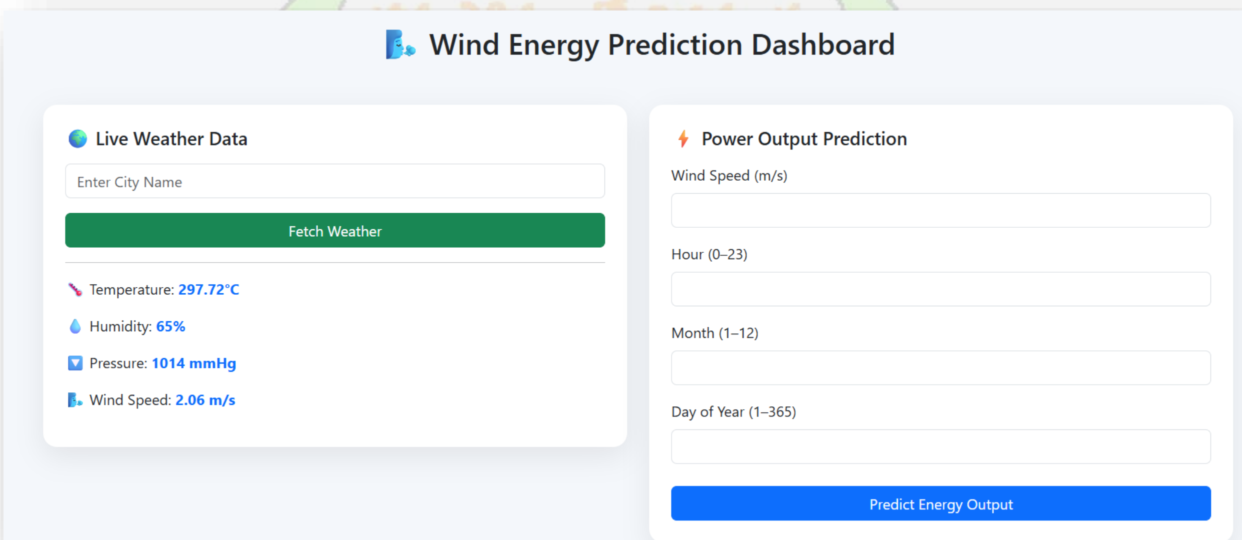
Prediction page:



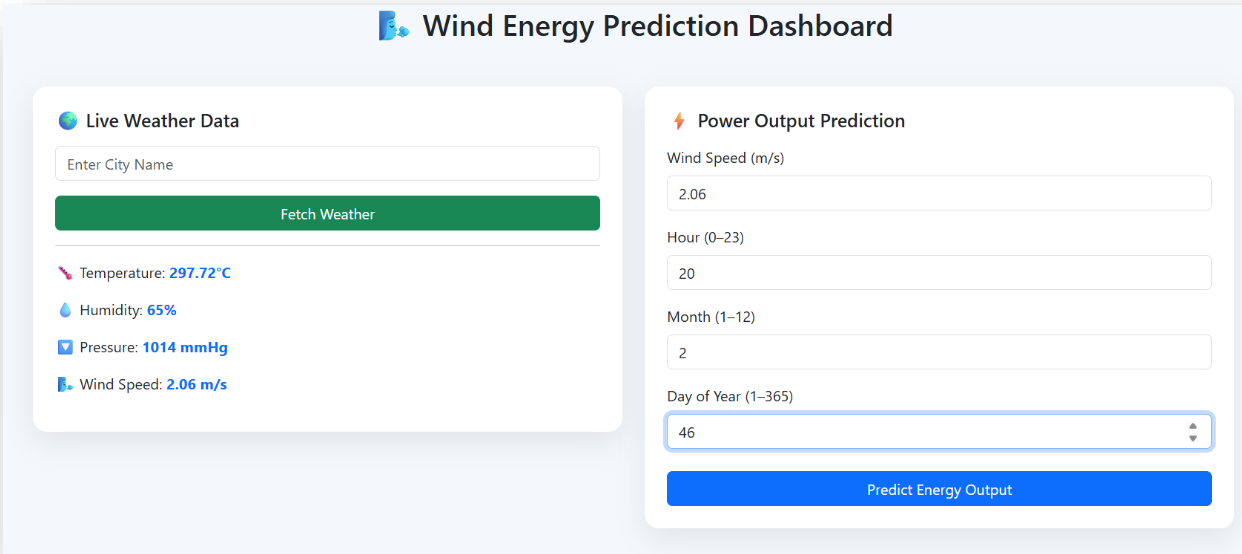
Input for 🌍 Live Weather Data:



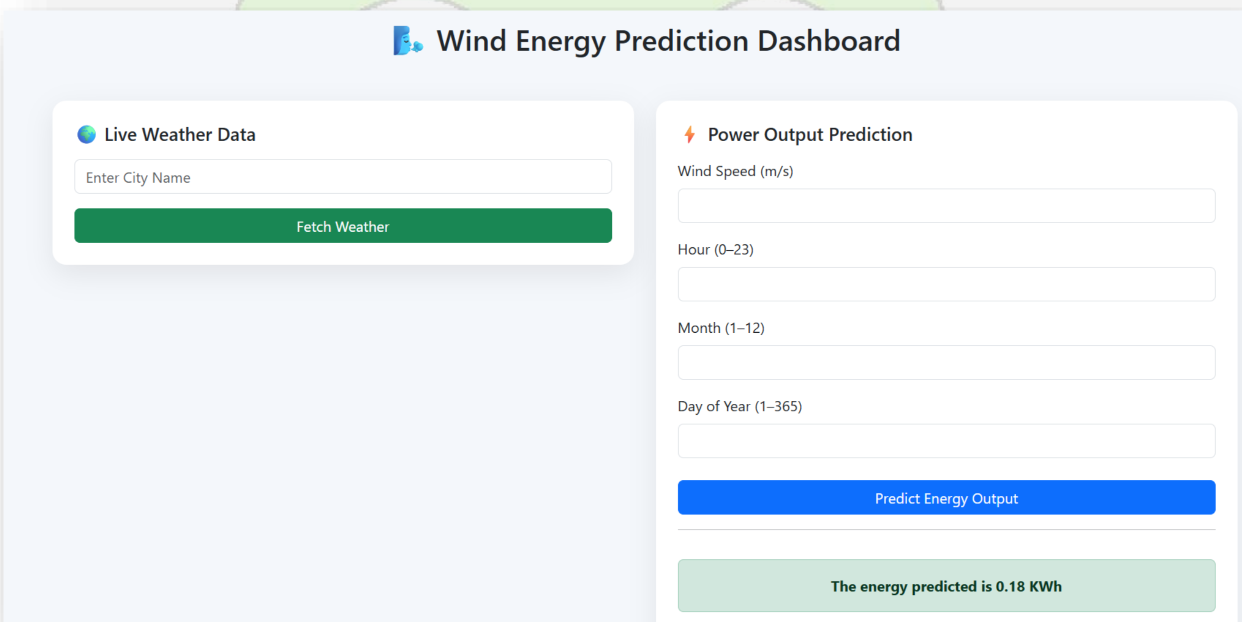
Output:



Input for ⚡ Power Output Prediction:



Output:



Chapter 9

Advantages and Disadvantages

9.1 – Advantages:

1. **Accurate Power Forecasting:**
   - Predicts turbine output using weather parameters (wind speed, direction, air density, temperature).
   - Reduces uncertainty in renewable energy generation.
   - Helps grid operators plan supply in advance.
2. **Improved Grid Stability:**
   - Prevents sudden power fluctuations.
   - Supports load balancing and demand–supply matching.
   - Enables smoother integration of renewable energy into the power grid.
3. **Higher Energy Efficiency:**
   - Optimizes turbine operation based on upcoming weather conditions.
   - Minimizes idle running and inefficient rotations.
   - Maximizes energy capture from available wind.
4. **Reduced Operational Cost:**
   - Avoids unnecessary turbine start/stop cycles.
   - Lowers maintenance expenses.
   - Reduces reliance on backup fossil-fuel generators.
5. **Predictive Maintenance:**
   - Detects abnormal patterns early.
   - Prevents equipment damage and breakdowns.
   - Extends turbine lifespan.

9.2 – Disadvantages:

1. Dependence on Weather Data Accuracy:

- Forecast quality relies heavily on meteorological data.
- Inaccurate or delayed weather inputs produce wrong power predictions.
- Sudden local weather changes are hard to capture.

2. Prediction Uncertainty:

- Wind is naturally chaotic and non-linear.
- Even advanced AI/ML models cannot guarantee 100% accuracy.
- Short-term turbulence and gusts reduce reliability.

3. High Initial Setup Cost:

- Requires sensors, data acquisition systems, storage servers, and software tools.
- Installation of weather monitoring equipment increases project budget.

4. Complex Model Development:

- Needs expertise in machine learning, data science, and power systems.
- Model tuning, feature engineering, and validation are time-consuming.
- Difficult for small organizations without technical teams.

5. Maintenance of Sensors & Equipment:

- Weather sensors degrade over time.
- Calibration and periodic replacement are necessary.
- Faulty sensors lead to misleading predictions.

6. Large Data Requirements:

- Requires long-term historical data for reliable training.
- Missing values and noisy data complicate preprocessing.

Chapter 10

Conclusion

10.1 – Conclusion:

The proposed weather-based wind turbine energy prediction system demonstrates how data-driven intelligence can significantly improve renewable energy utilization. By analysing meteorological parameters such as wind speed, temperature, direction, and air density, the model can accurately estimate power generation before actual production occurs. This enables operators and utilities to make proactive decisions instead of reactive adjustments.

The approach enhances grid reliability, optimizes turbine performance, and reduces operational costs by preventing unnecessary shutdowns and improving maintenance planning. It also supports efficient power scheduling and smoother integration of renewable energy into modern smart grids. As a result, dependence on fossil-fuel backup sources decreases, contributing to environmental sustainability and reduced carbon emissions.

Although challenges such as prediction uncertainty, data quality issues, and computational requirements exist, they can be minimized through better sensors, continuous model training, and advanced machine learning techniques. With further improvements and real-time data integration, the system can evolve into a fully automated renewable energy management solution.

Overall, weather-based wind energy prediction represents a crucial step toward intelligent, reliable, and sustainable power systems. It not only improves the efficiency of wind farms but also supports the global transition toward clean and green energy infrastructure.

Chapter 11

Future Scope

11.1 - Future Scope:

1. Real-Time Forecasting Systems:

- Integrate live weather APIs and on-site sensors for minute-level prediction.
- Enable instant turbine control adjustments based on incoming wind conditions.

2. Advanced AI & Deep Learning Models:

- Apply LSTM, GRU, and transformer-based models for better time-series forecasting.
- Improve prediction accuracy during sudden wind fluctuations and seasonal variations.

3. Integration with Smart Grids:

- Automatic coordination between wind farms, substations, and load demand.
- Dynamic power distribution to prevent overload and blackouts.

4. Energy Storage Optimization:

- Connect predictions with battery storage systems.
- Store excess power during high wind periods and release during low generation.

5. Hybrid Renewable Energy Systems:

- Combine wind prediction with solar forecasting models.
- Create reliable multi-source renewable energy plants.

6. Autonomous Turbine Control:

- Self-adjusting blade pitch and yaw angle using predicted wind direction and speed.
- Reduce mechanical stress and improve turbine lifespan.

Chapter 12

Appendix

12.1 - Source Code:

- **Intro.html:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Wind Energy Predictor</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boots
trap.min.css" rel="stylesheet">
    <style>
        body {
            background: linear-gradient(120deg, #0f2027, #203a43,
#2c5364);
            color: white;
        }
        .hero {
            height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
            text-align: center;
        }
        .btn-main {
            background: #00c6ff;
            border: none;
            color: black;
        }
    </style>
</head>
<body>
<div class="hero">
    <div>
        <h1 class="display-4 fw-bold">🌬️ Wind Power
Prediction</h1>
```

```
            <p class="lead mt-3">
                Machine Learning-based forecasting of wind turbine
energy output
            </p>
            <a href="/predict" class="btn btn-main btn-lg mt-4 px-5">
                Start Prediction
            </a>
        </div>
    </div>
</body>
</html>
```

Output:

- **Predict.html:**

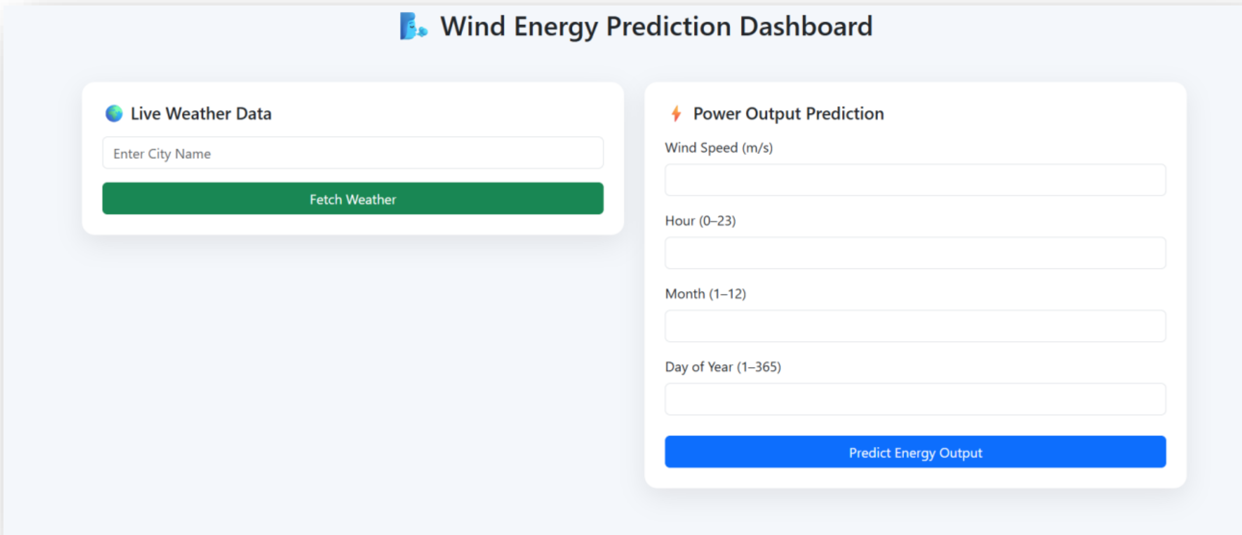
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Wind Power Dashboard</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boots
trap.min.css" rel="stylesheet">
    <style>
```

```
        body {
            background-color: #f4f7fb;
        }
        .card {
            border: none;
            border-radius: 15px;
            box-shadow: 0 10px 30px rgba(0,0,0,0.08);
        }
        .value {
            font-weight: bold;
            color: #0d6efd;
        }
    </style>
</head>
<body>
<div class="container mt-5">
    <h2  class="text-center  mb-5">🌬️  Wind  Energy  Prediction
Dashboard</h2>
    <div class="row">
        <!-- Weather Panel -->
        <div class="col-md-6 mb-4">
            <div class="card p-4">
                <h5 class="mb-3">🌍 Live Weather Data</h5>
                <form action="/windapi" method="POST">
                    <input  type="text"  name="city"  class="form-
control mb-3" placeholder="Enter City Name" required>
                    <button  class="btn  btn-success  w-100">Fetch
Weather</button>
                </form>

                {% if speed %}
                <hr>
                <p>🌡  Temperature:  <span  class="value">{{  temp
}}</span></p>
                <p>💧  Humidity:  <span  class="value">{{  humid
}}</span></p>
                <p>🔽  Pressure:  <span  class="value">{{  pressure
}}</span></p>
```

```
                <p>🌬️ Wind Speed: <span class="value">{{ speed
}}</span></p>
                {% endif %}
            </div>
        </div>
        <!-- Prediction Panel -->
        <div class="col-md-6 mb-4">
            <div class="card p-4">
                <h5 class="mb-3">⚡ Power Output Prediction</h5>
                <form action="/y_predict" method="POST">
                    <label class="form-label">Wind Speed
(m/s)</label>
                    <input type="number" step="any"
name="WindSpeed" class="form-control mb-3" required>
                    <label class="form-label">Hour (0–23)</label>
                    <input type="number" name="Hour" class="form-
control mb-3" required>
                    <label class="form-label">Month (1–
12)</label>
                    <input type="number" name="Month" class="form-
control mb-3" required>
                    <label class="form-label">Day of Year (1–
365)</label>
                    <input type="number" name="DayOfYear"
class="form-control mb-4" required>
                    <button class="btn btn-primary w-100">
                        Predict Energy Output
                    </button>
                </form>
                {% if prediction_text %}
                <hr>
                <div class="alert alert-success text-center mt-3">
                    <strong>{{ prediction_text }}</strong>
                </div>
                {% endif %}
            </div>
        </div>
    </div>
</div>
```

```
</body>
</html>
```

Output:



12.2 - Dataset Link:

- **Link:** https://www.kaggle.com/datasets/theforcecoder/wind-power-forecasting/code/data
- **QR Code:**

12.3 - GitHub and Project Demo Link:

- **GitHub link:**
  https://github.com/chitti4569/LTVIP2026TMIDS90356-KULALA-LIKHITHA
- **QR Code:**

- **Demo video Link:** https://drive.google.com/file/d/1hqQ5VPeipDQ0EkcxuaCpt4S2Y_mxj-Xs/view?usp=drive_link
- **QR Code:**