Predicting the energy output of wind turbine based on weather condition

A PROJECT REPORT

Submitted by

NAGARAJ.G - 711119104038

PALANI SELVAM. M - 711119104039

PASUPATHI DADEEJA - 711119104040

PUSHPAVENI.V - 711119104044

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

JANSONS INSTITUTE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

Project Report Format

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule
    c. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)
8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing
9. **RESULTS**
    a. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1.INTRODUCTION

1.1 PROJECT OVERVIEW

- Wind power generation differs from conventional thermal generation due to the stochastic nature of wind.

- Thus wind power forecasting plays a key role in dealing with the challenges of balancing supply and demand in any electricity system, given the uncertainty associated with the wind farm power output.

- Accurate wind power forecasting reduces the need for additional balancing energy and reserve power to integrate wind power.

- For a wind farm that converts wind energy into electricity power, a real-time prediction system of the output power is significant.

- In this guided project , a prediction system is developed with a method of combining statistical models and physical models.

- In this system, the inlet condition of the wind farm is forecasted by the auto regressive model.

1.2 PURPOSE

●Our aim is to map weather data to energy production.

●We wish to show that even data that is publicly available for weather stations close to wind farms can be used to give a good prediction of the energy output.

●Furthermore, we examine the impact of different weather conditions on the energy output of wind farms.

●We are building an IBM Watson AutoAI Machine Learning technique to predict the energy output of wind turbine.

●The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building.

●We are developing a web application which is built using node red service.

●We make use of the scoring end point to give user input values to the deployed model. The model prediction is then showcased on User Interface to predict the energy output

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Wind energy plays increasing role in the supply of energy world-wide. The energy-output of a wind farm is highly dependent on the weather condition present at its site. If the output is predicted more accurately, the energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, do energy prediction based on weather data and analyse the impo parameters as well as their correlation on the energy output.

To solve the wind power prediction problem, the Improved Sparrow Search
Algorithm-Least Squares Support Vector Machine (ISSA-LS-SVM) prediction
model based on chaotic sequences is proposed to improve the convergence accuracy and shorten the prediction time of the prediction model. Firstly, the problem in the historical data is decomposed using an ensemble empirical modal algorithm. Then, wind speed series prediction is performed using the LS-SVM model. Finally, the wind turbine output power prediction is performed. The results show that compared with LS-SVM, SSA-LS-SVM and Tent-SSA-LS-SVM models, the EEMD-ISSA LS-SVM prediction model has improved the convergence precision of wind power output predictive model, which is significant for the subsequent realization of optimal power dispatch.

Wind power into the electricity grid, precise and robust predictions are required. With increasing installed capacity and changing energy markets, there is a growing demand for short-term predictions.Machine learning methods can be used as a purely data-driven, spatio-temporal prediction model that yields better results than traditional physical models based on weather simulations. However, there are two big challenges when applying machine learning techniques to the domain of wind power predictions. First,when applying state of-the-art algorithms to big training data sets, the required computation times may increase to an unacceptable level. Second, the prediction performance and reliability have to be improved to cope with the requirements of the energy markets.

The prediction of wind power plays an indispensable role in maintaining the stability of the entire power grid. In this paper, a deep learning approach is proposed for the power prediction of multiple wind turbines. Starting from the time series of wind power, it is present a two-stage modeling strategy, in which a deep neural network combines spatiotemporal correlation to predict the power ofmultiple wind turbines. Specifically, the network is a joint model composed of Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN). Herein, the LSTM captures the temporal dependence of the historical power sequence, while the CNN extracts the spatial features among the data, thereby achieving the power prediction for multiple wind turbines. The proposed approach is validated by using the wind power data from an offshore wind farm , and the results in comparison with other

approaches shows the high prediction preciseness achieved.

2.2 REFERENCES

[1] Evolved Analytics LLC. DataModeler 8.0. Evolved Analytics LLC, 2010.

[2]A. M. Foleya, P. G. Leahya, A. Marvugliac, and E. J. McKeogha. Current methods and
advances in forecasting of wind power generation. Renewable Energy, 37:1–8, 2012.

[3] R. Jursa and K. Rohrig. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. International Journal of Forecasting, 24:694–709, 2008.

[4] M. Kotanchek, G. Smits, and E. Vladislavleva. Pursuing the pareto paradigm tournaments,algorithm variations & ordinal optimization. In Genetic Programming Theory and Prac-tice IV, volume 5 of Genetic and Evolutionary Computation, chapter 12, pages 167–186.
Springer, 11-13 May 2006.

[5] J. R. Koza. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT
Press, Cambridge Massachusetts, May 1994.

[6] O. Kramer and F. Gieseke. Analysis of wind energy time series with kernel methods and
neural networks. In Seventh International Conference on Natural Computation, 2011. to appear.

[7] O. Kramer and F. Gieseke. Short-term wind energy forecasting using support vector regression. In International Conference on Soft Computing Models in Industrial and Environmental Applications, pages 271–280. Springer, 2011.

[8] A. Kusiak, H. Zheng, and Z. Song. Short-term prediction of wind farm power: A data mining approach. IEEE Transactions on Energy Conversion, 24(1):125 – 136, 2009.

[9] R. Poli, W. B. Langdon, and N. F. McPhee. A Field Guide to Genetic Programming.
lulu.com, 2008.

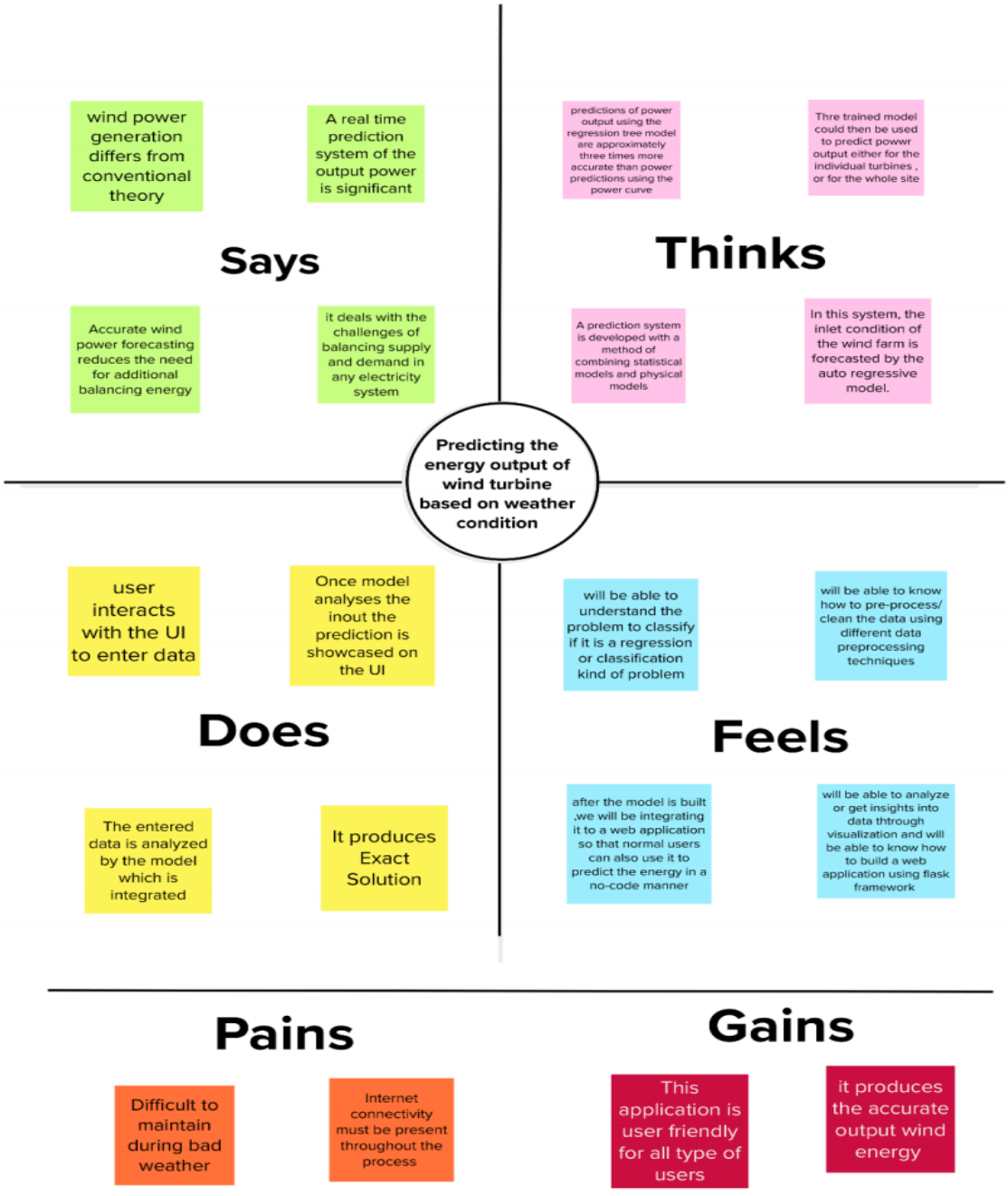
[10] M. Schmidt and H. Lipson. Age-fitness pareto optimization. In Genetic Programming Theory and Practice.

2.3 PROBLEM STATEMENT DEFNITION

It is very difficult to predict the energy output of wind turbine. To avoid the difficulties they can use a web application for predicting the output of wind turbine. The energy output of the wind turbine is given in the web application based on weather condition.

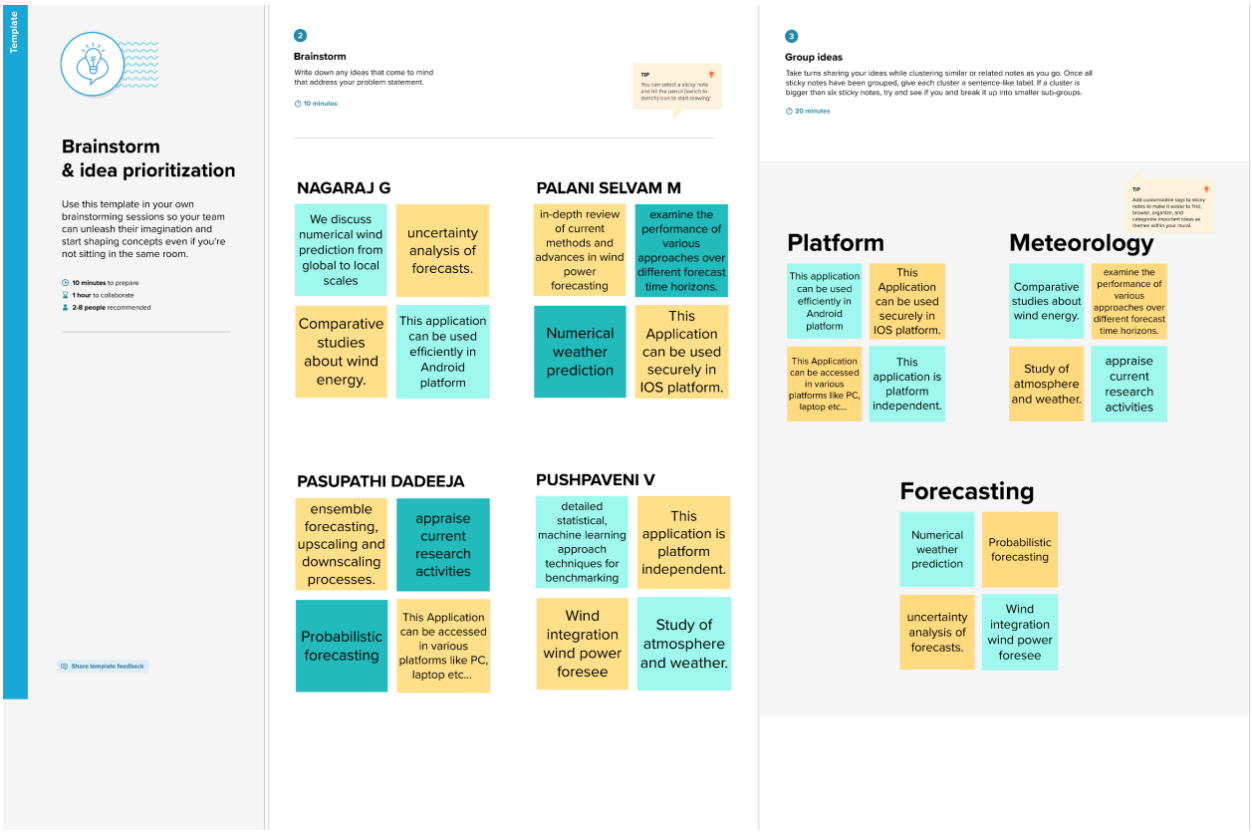
3.IDEATION AND PROPOSED SOLUTION

3. EMPATHY MAP CANVAS

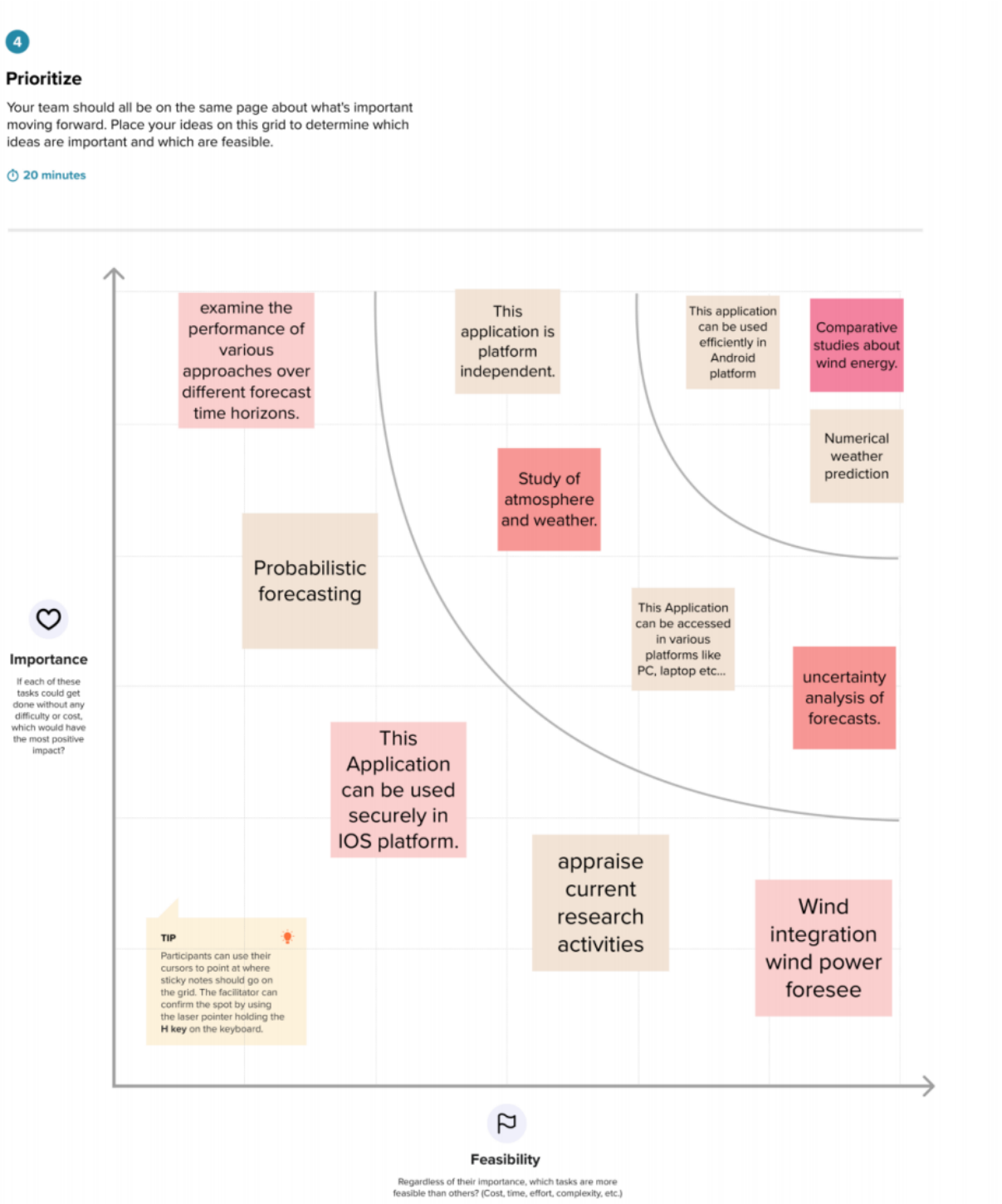


3.2 IDEATION AND BRAINSTROMING

Step-1: Brainstorm, Idea Listing and Grouping



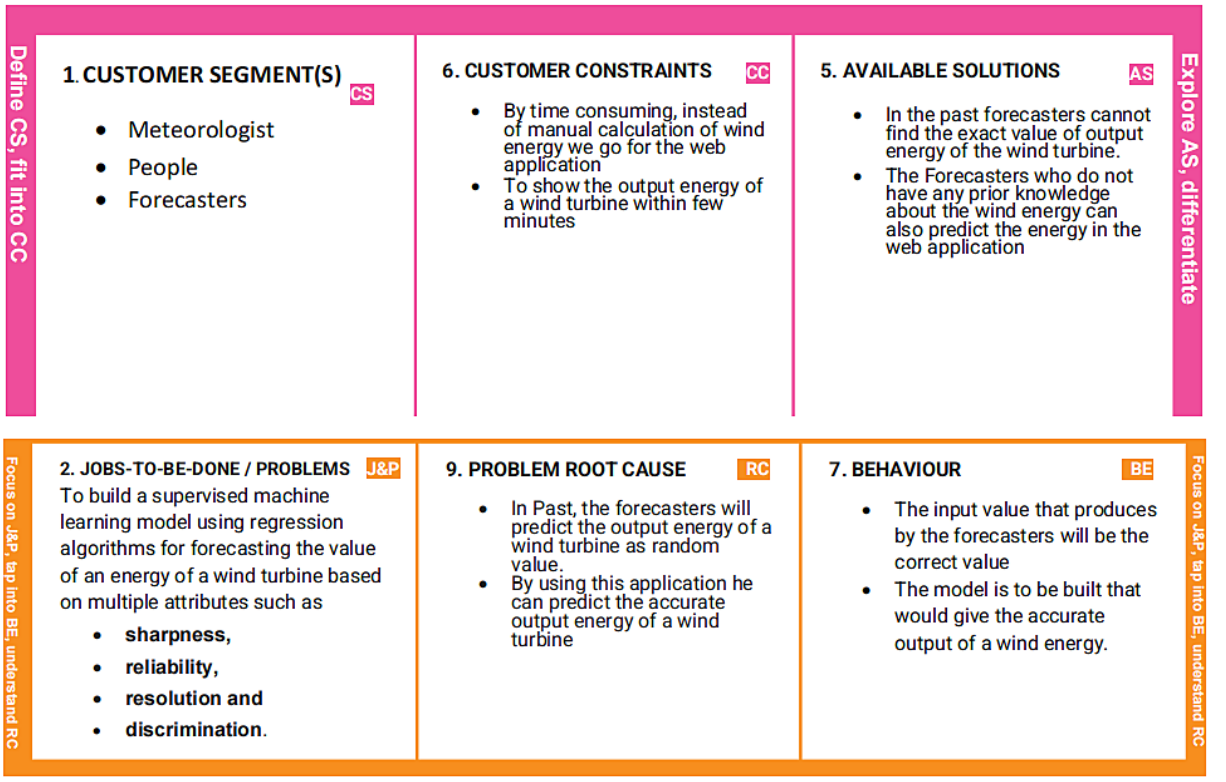
Step-2: Idea Prioritization



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The main aim of this project is to predict the accurate output energy of a wind turbine based on weather condition using the Machine Learning (ML) algorithms and collection of data about the input values of wind energy. The project should take parameters related to wind speed, wind direction, temperature, pressure, and humidity as inputs.
2.	Idea / Solution description	By using this model, The Energy Output of a wind turbine can predict accurately.
3.	Novelty / Uniqueness	The accuracy of the prediction by using the input values such as wind speed, wind direction, temperature, pressure, and humidity as inputs.
4.	Social Impact / Customer Satisfaction	If the meteorologist wants to know the energy that produces by the wind, It predicts the exact output to the forecasters or the meteorologist.
5.	Business Model (Revenue Model)	It helps meteorologist and the forecasters to predict the accurate output by remotely without any incorrect values.
6.	Scalability of the Solution	Using Stored data and machine learning approaches, this project proposed a scalable framework for predicting values of wind energy for different weather condition.

3.4 PROBLEM FIT SOLUTION



4.REQUIREMENT ANALYSIS

4.4 FUNCTIONAL REQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login through (web)	Login registered mail id and password
FR-4	User login through (mobile app)	Login registered mail id and password
FR-5	User needs	In the app, enter your input values.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system should be user friendly for the forecaster. It should recommend predicting the wind energy.
NFR-2	**Security**	The login information should not be accessed by any other user than the respective. The user data should be private.
NFR-3	**Reliability**	The Forecaster can be able to guess the approximate value of the wind energy.
NFR-4	**Performance**	The wind energy value will get updated every time consecutively according to the weather condition.
NFR-5	**Availability**	It can be used from anywhere at any time.
NFR-6	**Scalability**	It is easily adaptable. This device can handle any number of registrations.

5. PROJECT DESIGN

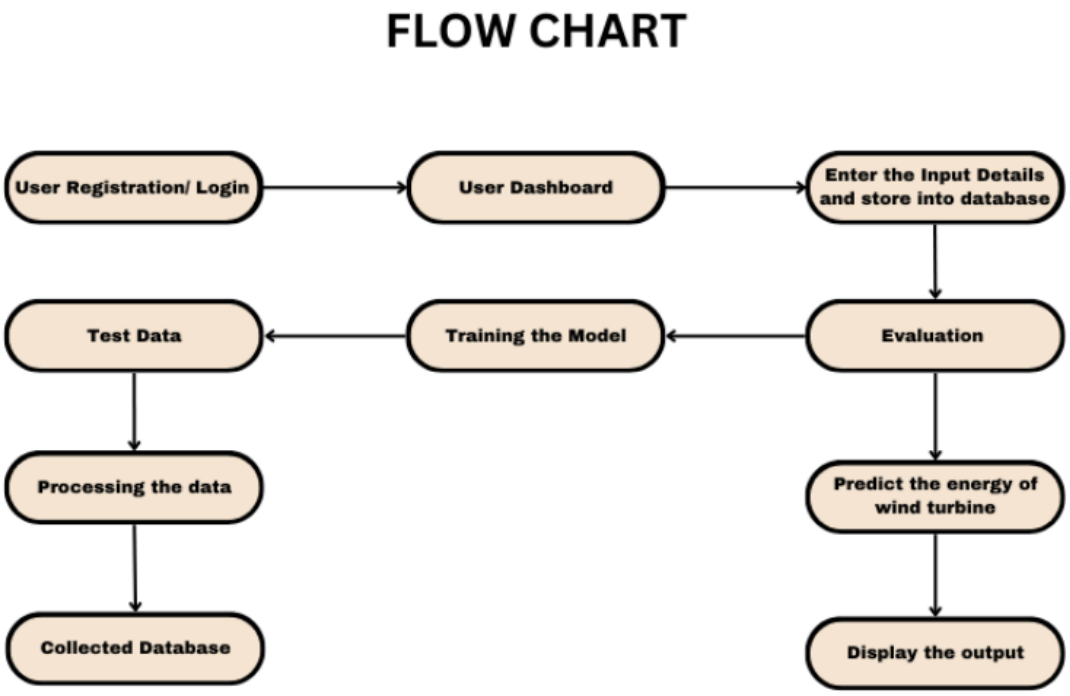
5.1 DATA FLOW DIAGRAMS

1, The user should register on web application using mail ID and password.

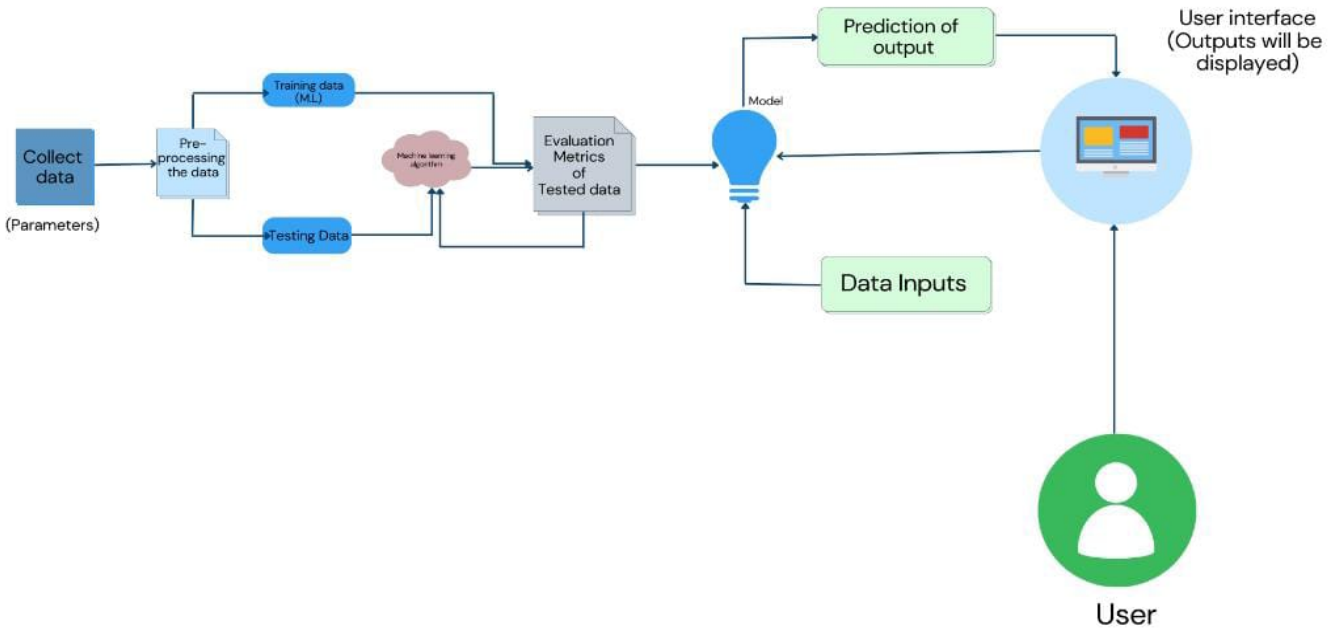
2. The user should enter the wind details such as wind speed, wind direction and temperature.

3. The data given by the forecaster is taken as the input and will get stored in IBM cloud database.

4. The application is build to predict the energy that produces by wind turbine based on weather condition.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

Use the below template to list all the user stories for the product,

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin	Data set	USN-1	Gather the information needed to make the wind energy prediction value	Enough data has been gathered to train the model	High	Sprint-1
	Data Pre-processing	USN-2	Perform data cleaning to optimize the dataset	Clean Dataset enough to make accurate predictions	High	Sprint-1
	Training & Building Model	USN-3	Building the model using linear regression algorithms to classify the data	Model should be used for predicting exact valuation of the wind energy.	High	Sprint-1
	Deploy the model	USN-4	Deployment of ML model using IBM Cloud	Model should be working fine from the cloud	High	Sprint-2
	Integrate the web application with the IBM model	USN-5	Use flask for the integration purpose	The model ought to be simple to use and reliable on the web application.	High	Sprint-2
Meteorologist	Home page	USN-6	Information on the application and the process for wind energy prediction.	We can get an idea about how to use these model	Medium	Sprint-2
	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password	User can access my account / dashboard	High	Sprint-3
	Login	USN-8	As a user, I can log into the application by entering email & password	User can login to my account	High	Sprint-3

	Dashboard	USN-9	User can enter the details and get the prediction values.	User can add input values	Medium	Sprint-4
	Output	USN-10	As a user, I can be able to access the predicted output value of wind energy	Predicted values must be displayed depending on the data provided by the user.	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection and Preprocessing	USN-1	As a user, I am unable to engage with anything	9	Medium	Nagaraj G Palani Selvam M Pasupathi Dadeeja
Sprint-2	Performance Testing	USN-2	As a user, I am unable to engage with anything	6	High	Nagaraj G Palani Selvam M Pasupathi Dadeeja
Sprint-2	Execute and Test your Model	USN-3	As a user, I can predict the wind energy using the best created ML models	5	High	Nagaraj G Palani Selvam M Pasupathi Dadeeja
Sprint-3	Train the ML model	USN-4	As a user, I can predict wind energy using the user interface	6	Medium	Nagaraj G Palani Selvam M Pasupathi Dadeeja
Sprint-3	Integrate Flask with Model	USN-5	As a user, I can predict the wind energy using the best created ML models	5	High	Nagaraj G Palani Selvam M Pasupathi Dadeeja
Sprint-4	Model Deployment on IBM cloud using IBM watson	USN-6	As a user, I can use the model by requesting the deployed model on cloud	9	High	Nagaraj G Palani Selvam M Pasupathi Dadeeja

6.2. Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	9	4 Days	24 Oct 2022	27 Oct 2022	9	28 Oct 2022
Sprint-2	11	6 Days	31 Oct 2022	05 Nov 2022	11	06 Nov 2022
Sprint-3	11	5 Days	07 Nov 2022	11 Nov 2022	11	12 Nov 2022
Sprint-4	9	5 Days	14 Nov 2022	18 Nov 2022	9	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

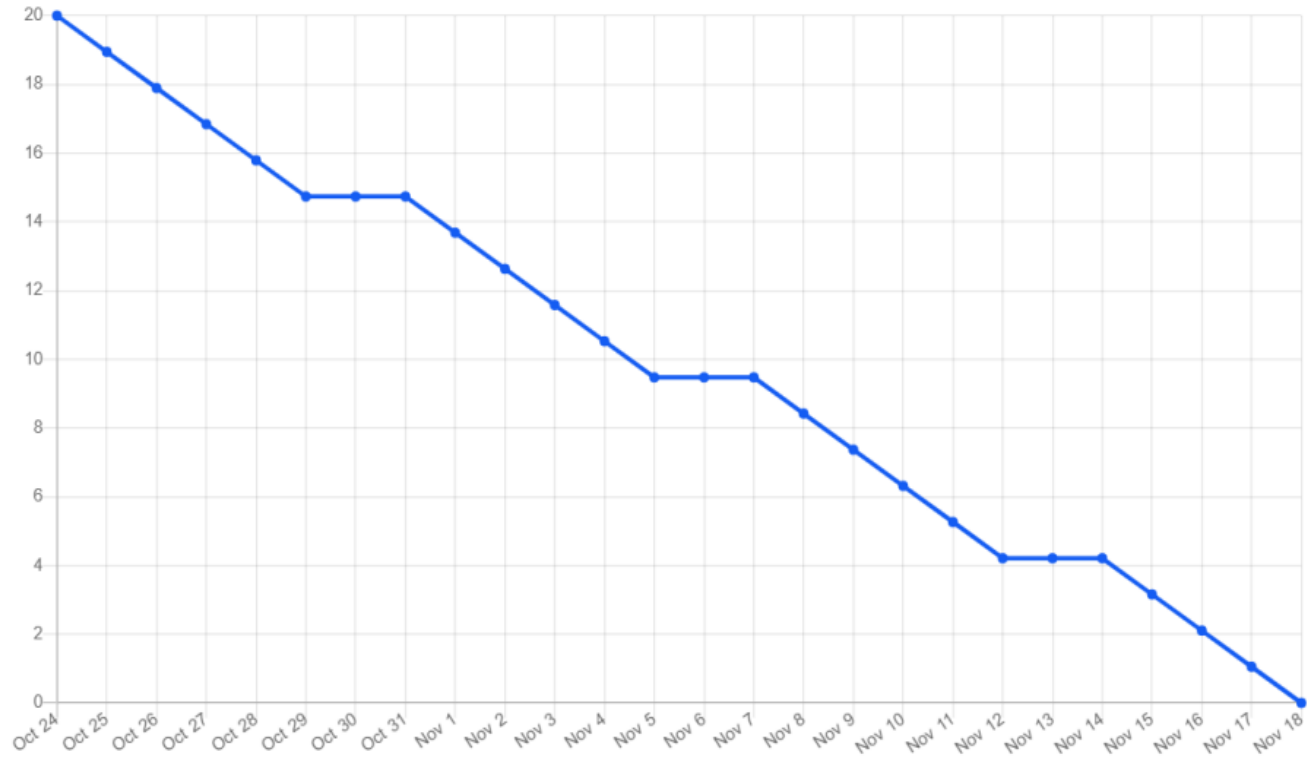
$$AV = \frac{sprint\ duration}{velocity}$$

$$= 9+11+11+9/20 = 2.0$$

Burndown Chart:

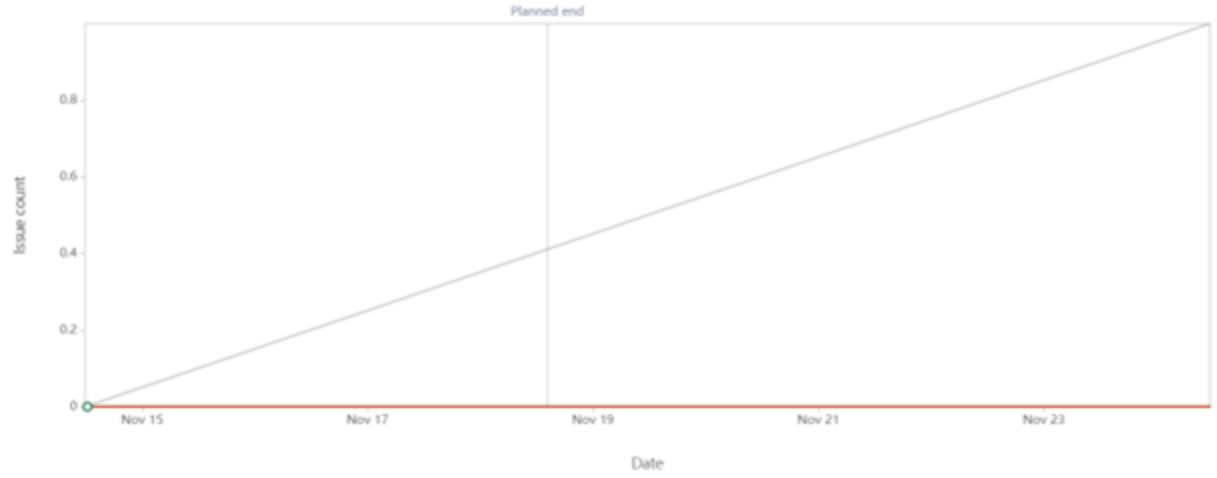
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Burndown Chart

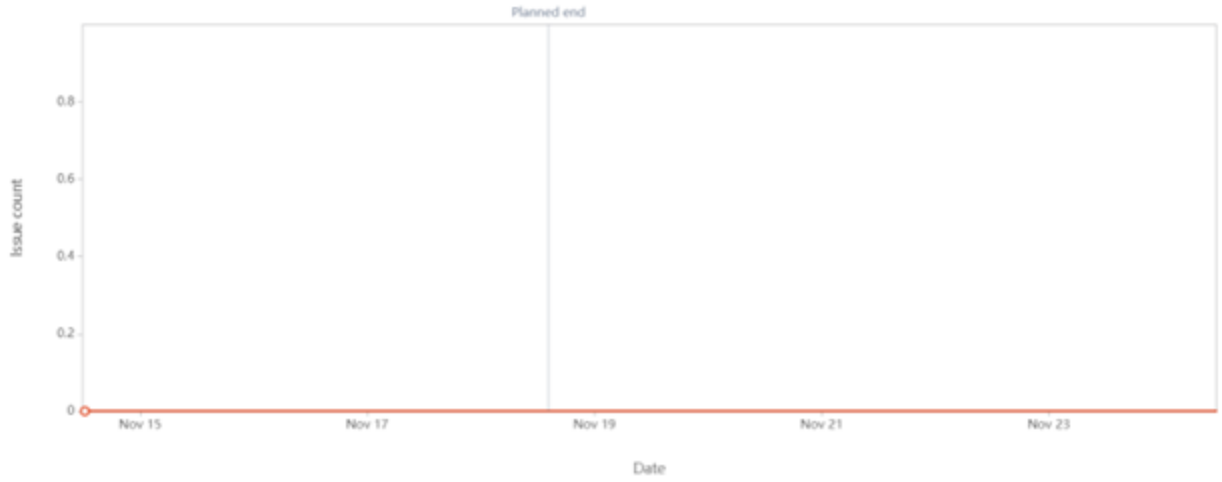


6.3 Report from JIRA

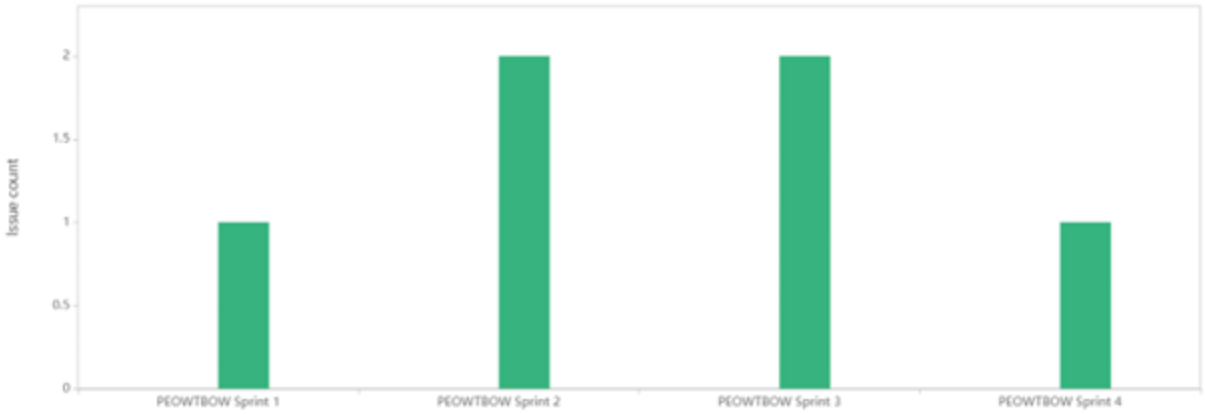
Burnup report



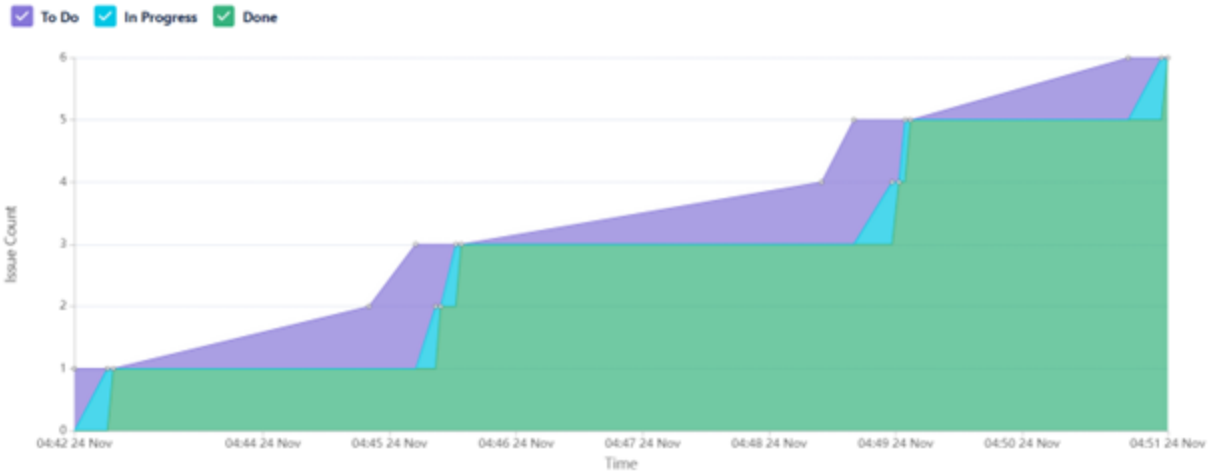
Sprint burndown chart



Velocity report



Cumulative flow diagram



7. Coding and Solutioning

7.1 Feature - 1:

```
<html>
<head>
    <meta charset="utf-8">
    <title>Predicting the Energy Output of a Wind Turbine Based on Weather Condition</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta content="" name="keywords">
    <meta content="" name="description">

    <!-- Favicon -->
    <link href="img/favicon.ico" rel="icon">

    <!-- Google Web Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;500&family=Roboto:wg
ht@500;700;900&display=swap" rel="stylesheet">

    <!-- Icon Font Stylesheet -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">

    <!-- Libraries Stylesheet -->
    <link href="../static/lib/animate/animate.min.css" rel="stylesheet">
    <link href="../static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
```

```
    <link href="../static/lib/lightbox/css/lightbox.min.css" rel="stylesheet">


    <!-- Customized Bootstrap Stylesheet -->
    <link href="../static/css/bootstrap.min.css" rel="stylesheet">


 <!-- Template Stylesheet -->
    <link href="../static/css/style.css" rel="stylesheet">
</head>


<body>
    <!-- Spinner Start -->
    <div id="spinner" class="show bg-white position-fixed translate-middle w-100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
        <div class="spinner-border text-primary" style="width: 3rem; height: 3rem;" role="status">
            <span class="sr-only">Loading...</span>
        </div>
    </div>
    <!-- Spinner End -->



    <!-- Topbar Start -->
    <div class="container-fluid bg-dark p-0">
        <div class="row gx-0 d-none d-lg-flex">
            <div class="col-lg-7 px-5 text-start">
                <div class="h-100 d-inline-flex align-items-center me-4">
                    <small class="fa fa-map-marker-alt text-primary me-2"></small>
                    <small>Tamilnadu, India</small>
                </div>
```

```
            <!--<div class="h-100 d-inline-flex align-items-center">
                <small class="far fa-clock text-primary me-2"></small>
                <small>Mon - Fri : 09.00 AM - 09.00 PM</small>
            </div>-->
        </div>
        <div class="col-lg-5 px-5 text-end">
            <div class="h-100 d-inline-flex align-items-center me-4">
                <small class="fa fa-phone-alt text-primary me-2"></small>
                <small>+012 345 6789</small>
            </div>
            <div class="h-100 d-inline-flex align-items-center mx-n2">
                <a class="btn btn-square btn-link rounded-0 border-0 border-end border-secondary" href=""><i class="fab fa-facebook-f"></i></a>
                <a class="btn btn-square btn-link rounded-0 border-0 border-end border-secondary" href=""><i class="fab fa-twitter"></i></a>
                <a class="btn btn-square btn-link rounded-0 border-0 border-end border-secondary" href=""><i class="fab fa-linkedin-in"></i></a>
                <a class="btn btn-square btn-link rounded-0" href=""><i class="fab fa-instagram"></i></a>
            </div>
        </div>
    </div>
    <!-- Topbar End -->


    <!-- Navbar Start -->
    <nav class="navbar navbar-expand-lg bg-white navbar-light sticky-top p-0">
        <a href="index.html" class="navbar-brand d-flex align-items-center border-end px-4 px-lg-5">
            <h2 class="m-0 text-primary">Wind Turbine</h2>
```

```
        </a>
        <button type="button" class="navbar-toggler me-4" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <div class="navbar-nav ms-auto p-4 p-lg-0">
                <a href="/" class="nav-item nav-link active">Home</a>
                <a href="/login" class="nav-item nav-link">Admin</a>
                <!--<div class="nav-item dropdown">
                    <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Pages</a>
                    <div class="dropdown-menu bg-light m-0">
                        <a href="feature.html" class="dropdown-item">Feature</a>
                        <a href="quote.html" class="dropdown-item">Free Quote</a>
                        <a href="team.html" class="dropdown-item">Our Team</a>
                        <a href="testimonial.html" class="dropdown-item">Testimonial</a>
                        <a href="404.html" class="dropdown-item">404 Page</a>
                    </div>
                </div>-->
            </div>
            <a href="/login_user" class="btn btn-primary rounded-0 py-4 px-lg-5 d-none d-lg-block">Test<i class="fa fa-arrow-right ms-3"></i></a>
        </div>
    </nav>
    <!-- Navbar End -->


    <!-- Carousel Start -->
    <div class="container-fluid p-0 pb-5 wow fadeIn" data-wow-delay="0.1s">
```

```
<div class="owl-carousel header-carousel position-relative">
                        <div class="owl-carousel-item position-relative" data-dot="<img src='../static/img/wn4.jpg'>">
            <img class="img-fluid" src="../static/img/wn4.jpg" alt="">

            <div class="owl-carousel-inner">
              <div class="container">
                <div class="row justify-content-start">
                  <div class="col-10 col-lg-8">
                    <h1 class="display-2 text-white animated slideInDown">Wind Power Prediction</h1>
                    <p class="fs-5 fw-medium text-white mb-4 pb-3">Predicting the Energy Output of a Wind Turbine Based on Weather Condition</p>
                    <a href="" class="btn btn-primary rounded-pill py-3 px-5 animated slideInLeft">Read More</a>
                  </div>
                </div>
              </div>
            </div>
          <div class="owl-carousel-item position-relative" data-dot="<img src='../static/img/carousel-1.jpg'>">
            <img class="img-fluid" src="../static/img/carousel-1.jpg" alt="">
            <div class="owl-carousel-inner">
              <div class="container">
                <div class="row justify-content-start">
                  <div class="col-10 col-lg-8">
                    <h1 class="display-2 text-white animated slideInDown">Pioneers Of Solar And Renewable Energy</h1>
                    <p class="fs-5 fw-medium text-white mb-4 pb-3">Vero elitr justo clita lorem. Ipsum dolor at sed stet sit diam no. Kasd rebum ipsum et diam justo clita et kasd rebum sea
```

```
elitr.</p>
                        <a href="" class="btn btn-primary rounded-pill py-3 px-5 animated slideInLeft">Read More</a>
                    </div>
                </div>
            </div>
        </div>
        <div class="owl-carousel-item position-relative" data-dot="<img src='../static/img/carousel-3.jpg'>">
            <img class="img-fluid" src="../static/img/carousel-3.jpg" alt="">
            <div class="owl-carousel-inner">
                <div class="container">
                    <div class="row justify-content-start">
                        <div class="col-10 col-lg-8">
                            <h1 class="display-2 text-white animated slideInDown">Pioneers Of Solar And Renewable Energy</h1>
                            <p class="fs-5 fw-medium text-white mb-4 pb-3">Vero elitr justo clita lorem. Ipsum dolor at sed stet sit diam no. Kasd rebum ipsum et diam justo clita et kasd rebum sea elitr.</p>
                            <a href="" class="btn btn-primary rounded-pill py-3 px-5 animated slideInLeft">Read More</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <!-- Carousel End -->
```

```
    <!-- Feature Start -->


    <!-- Feature Start -->



    <!-- About Start -->


    <!-- About End -->



    <!-- Service Start -->
    <div class="container-xxl py-5">
        <div class="container">
            <div class="text-center mx-auto mb-5 wow fadeInUp" data-wow-delay="0.1s" style="max-width: 600px;">
                <h6 class="text-primary">Our Services</h6>
                <h1 class="mb-4">Predicting the Energy Output of a Wind Turbine Based on Weather Condition</h1>
            </div>
            <div class="row g-4">
                <div class="col-md-6 col-lg-4 wow fadeInUp" data-wow-delay="0.1s">
                    <div class="service-item rounded overflow-hidden">
                        <img class="img-fluid" src="../static/img/wn6.jpg" alt="">
                        <div class="position-relative p-4 pt-0">
                            <div class="service-icon">
                                <i class="fa fa-solar-panel fa-3x"></i>
                            </div>
```

```
                <h4 class="mb-3">Wind Turbines</h4>
                <p>Stet stet justo dolor sed duo. Ut clita sea sit ipsum diam lorem diam.</p>
                <a class="small fw-medium" href="">Read More<i class="fa fa-arrow-right ms-2"></i></a>
            </div>
        </div>
    </div>
    <div class="col-md-6 col-lg-4 wow fadeInUp" data-wow-delay="0.3s">
        <div class="service-item rounded overflow-hidden">
            <img class="img-fluid" src="../static/img/img-600x400-2.jpg" alt="">
            <div class="position-relative p-4 pt-0">
                <div class="service-icon">
                    <i class="fa fa-wind fa-3x"></i>
                </div>
                <h4 class="mb-3">Wind Turbines</h4>
                <p>Stet stet justo dolor sed duo. Ut clita sea sit ipsum diam lorem diam.</p>
                <a class="small fw-medium" href="">Read More<i class="fa fa-arrow-right ms-2"></i></a>
            </div>
        </div>
    </div>
    <div class="col-md-6 col-lg-4 wow fadeInUp" data-wow-delay="0.5s">
        <div class="service-item rounded overflow-hidden">
            <img class="img-fluid" src="../static/img/wn5.jpg" alt="">
            <div class="position-relative p-4 pt-0">
                <div class="service-icon">
                    <i class="fa fa-lightbulb fa-3x"></i>
                </div>
                <h4 class="mb-3">Wind Turbines</h4>
```

```
                    <p>Stet stet justo dolor sed duo. Ut clita sea sit ipsum diam lorem diam.</p>
                    <a class="small fw-medium" href="">Read More<i class="fa fa-arrow-right ms-
2"></i></a>
                </div>
            </div>
        </div>

    </div>
</div>
<!-- Service End -->


<!-- Feature Start -->

<!-- Feature End -->


<!-- Projects Start -->

<!-- Projects End -->


<!-- Quote Start -->
<div class="container-fluid bg-light overflow-hidden my-5 px-lg-0">
    <div class="container quote px-lg-0">
        <div class="row g-0 mx-lg-0">
            <div class="col-lg-6 ps-lg-0 wow fadeIn" data-wow-delay="0.1s" style="min-height:
400px;">
```

```
            <div class="position-relative h-100">
                <img class="position-absolute img-fluid w-100 h-100" src="img/quote.jpg" style="object-fit: cover;" alt="">
            </div>
        </div>
        <div class="col-lg-6 quote-text py-5 wow fadeIn" data-wow-delay="0.5s">
            <div class="p-lg-5 pe-lg-0">
                <h6 class="text-primary">Free Quote</h6>
                <h1 class="mb-4">Get A Free Quote</h1>
                <p class="mb-4 pb-2">Tempor erat elitr rebum at clita. Diam dolor diam ipsum sit. Aliqu diam amet diam et eos. Clita erat ipsum et lorem et sit, sed stet lorem sit clita duo justo erat amet</p>
                <form>
                    <div class="row g-3">
                        <div class="col-12 col-sm-6">
                            <input type="text" class="form-control border-0" placeholder="Your Name" style="height: 55px;">
                        </div>
                        <div class="col-12 col-sm-6">
                            <input type="email" class="form-control border-0" placeholder="Your Email" style="height: 55px;">
                        </div>
                        <div class="col-12 col-sm-6">
                            <input type="text" class="form-control border-0" placeholder="Your Mobile" style="height: 55px;">
                        </div>
                        <div class="col-12 col-sm-6">
                            <select class="form-select border-0" style="height: 55px;">
                                <option selected>Select A Service</option>
                                <option value="1">Service 1</option>
```

```
                        <option value="2">Service 2</option>
                        <option value="3">Service 3</option>
                    </select>
                </div>
                <div class="col-12">
                    <textarea class="form-control border-0" placeholder="Special
Note"></textarea>
                </div>
                <div class="col-12">
                    <button class="btn btn-primary rounded-pill py-3 px-5"
type="submit">Submit</button>
                </div>
            </div>
        </form>
    </div>
</div>
</div>
</div>
<!-- Quote End -->


<!-- Team Start -->


<!-- Team End -->


<!-- Testimonial Start -->
```

```
    <!-- Testimonial End -->



    <!-- Footer Start -->
    <div class="container-fluid bg-dark text-body footer mt-5 pt-5 wow fadeIn" data-wow-delay="0.1s">
        <div class="container py-5">
            <div class="row g-5">
                <div class="col-lg-3 col-md-6">
                    <h5 class="text-white mb-4">Address</h5>
                    <p class="mb-2"><i class="fa fa-map-marker-alt me-3"></i>Tamilnadu, India</p>
                    <p class="mb-2"><i class="fa fa-phone-alt me-3"></i>+012 345 67890</p>
                    <p class="mb-2"><i class="fa fa-envelope me-3"></i>windturbine@info.com</p>
                    <div class="d-flex pt-2">
                        <a class="btn btn-square btn-outline-light btn-social" href=""><i class="fab fa-twitter"></i></a>
                        <a class="btn btn-square btn-outline-light btn-social" href=""><i class="fab fa-facebook-f"></i></a>
                        <a class="btn btn-square btn-outline-light btn-social" href=""><i class="fab fa-youtube"></i></a>
                        <a class="btn btn-square btn-outline-light btn-social" href=""><i class="fab fa-linkedin-in"></i></a>
                    </div>
                </div>
                <div class="col-lg-3 col-md-6">
                    <h5 class="text-white mb-4">Quick Links</h5>
                    <a class="btn btn-link" href="/">Home</a>
                    <a class="btn btn-link" href="/login">Admin</a>
                    <a class="btn btn-link" href="/login_user">Test</a>
                </div>
```

```
        </div>
      </div>
      <div class="container">
        <div class="copyright">
          <div class="row">
            <div class="col-md-6 text-center text-md-start mb-3 mb-md-0">
              Wind Turbine <a href="#"></a>
            </div>
            <div class="col-md-6 text-center text-md-end">
              <!--/*** This template is free as long as you keep the footer author’s credit link/attribution link/backlink. If you'd like to use the template without the footer author’s credit link/attribution link/backlink, you can purchase the Credit Removal License from "https://htmlcodex.com/credit-removal". Thank you for your support. ***/-->
               <a href="https://htmlcodex.com"></a>
              <br> <a href="https://themewagon.com" target="_blank"></a>
            </div>
          </div>
        </div>
      </div>
    </div>
    <!-- Footer End -->


    <!-- Back to Top -->
    <a href="#" class="btn btn-lg btn-primary btn-lg-square rounded-circle back-to-top"><i class="bi bi-arrow-up"></i></a>
```

```
    <!-- JavaScript Libraries -->
    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></script>
    <script src="../static/lib/wow/wow.min.js"></script>
    <script src="../static/lib/easing/easing.min.js"></script>
    <script src="../static/lib/waypoints/waypoints.min.js"></script>
    <script src="../static/lib/counterup/counterup.min.js"></script>
    <script src="../static/lib/owlcarousel/owl.carousel.min.js"></script>
    <script src="../static/lib/isotope/isotope.pkgd.min.js"></script>
    <script src="../static/lib/lightbox/js/lightbox.min.js"></script>

    <!-- Template Javascript -->
    <script src="../static/js/main.js"></script>
</body>

</html>
```

7.2 Feature 2 :

Testing

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


pd.pandas.set_option('display.max_columns', None)
```

```
dataset = pd.read_csv('static/dataset/train.csv')
dat=dataset.head()
print(dat)
```

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import seaborn as sns
import matplotlib.pyplot as plt
#%matplotlib inline

from sklearn.metrics import r2_score
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os

from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error
```

```
train = pd.read_csv('static/dataset/train.csv')
test = pd.read_csv('static/dataset/test.csv')
train.head()
test.head()
train.describe()

test.describe()
##
corr = train.corr()
plt.figure(figsize=(20,10))
mask = np.zeros_like(corr,dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
#sns.heatmap(corr,mask=mask,annot=True)
#plt.show()

###
train.drop(['generator_temperature(°C)','windmill_body_temperature(°C)'],inplace=True,axis=1)
test.drop(['generator_temperature(°C)','windmill_body_temperature(°C)'],inplace=True,axis=1)

train.isnull().sum()
test.isnull().sum()
#sns.heatmap(train.isnull(),cbar=False,yticklabels=False,cmap = 'viridis')
#sns.heatmap(test.isnull(),cbar=False,yticklabels=False,cmap = 'viridis')



train['gearbox_temperature(°C)'].fillna(train['gearbox_temperature(°C)'].mean(),inplace=True)
train['area_temperature(°C)'].fillna(train['area_temperature(°C)'].mean(),inplace=True)
```

```
train['rotor_torque(N-m)'].fillna(train['rotor_torque(N-m)'].mean(),inplace=True)

train['blade_length(m)'].fillna(train['blade_length(m)'].mean(),inplace=True)

train['blade_breadth(m)'].fillna(train['blade_breadth(m)'].mean(),inplace=True)

train['windmill_height(m)'].fillna(train['windmill_height(m)'].mean(),inplace=True)

train['cloud_level'].fillna(train['cloud_level'].mode()[0],inplace=True)

train['atmospheric_temperature(°C)'].fillna(train['atmospheric_temperature(°C)'].mean(),inplace=True)

train['atmospheric_pressure(Pascal)'].fillna(train['atmospheric_pressure(Pascal)'].mean(),inplace=True)

train['wind_speed(m/s)'].fillna(train['wind_speed(m/s)'].mean(),inplace=True)

train['shaft_temperature(°C)'].fillna(train['shaft_temperature(°C)'].mean(),inplace=True)

train['blades_angle(°)'].fillna(train['blades_angle(°)'].mean(),inplace=True)

train['engine_temperature(°C)'].fillna(train['engine_temperature(°C)'].mean(),inplace=True)

train['motor_torque(N-m)'].fillna(train['motor_torque(N-m)'].mean(),inplace=True)

train['wind_direction(°)'].fillna(train['wind_direction(°)'].mean(),inplace=True)


test['gearbox_temperature(°C)'].fillna(test['gearbox_temperature(°C)'].mean(),inplace=True)

test['area_temperature(°C)'].fillna(test['area_temperature(°C)'].mean(),inplace=True)

test['rotor_torque(N-m)'].fillna(test['rotor_torque(N-m)'].mean(),inplace=True)

test['blade_length(m)'].fillna(test['blade_length(m)'].mean(),inplace=True)

test['blade_breadth(m)'].fillna(test['blade_breadth(m)'].mean(),inplace=True)

test['windmill_height(m)'].fillna(test['windmill_height(m)'].mean(),inplace=True)

test['cloud_level'].fillna(test['cloud_level'].mode()[0],inplace=True)

test['atmospheric_temperature(°C)'].fillna(test['atmospheric_temperature(°C)'].mean(),inplace=True)

test['atmospheric_pressure(Pascal)'].fillna(test['atmospheric_pressure(Pascal)'].mean(),inplace=True)

test['wind_speed(m/s)'].fillna(test['wind_speed(m/s)'].mean(),inplace=True)

test['shaft_temperature(°C)'].fillna(test['shaft_temperature(°C)'].mean(),inplace=True)

test['blades_angle(°)'].fillna(test['blades_angle(°)'].mean(),inplace=True)
```

```
test['engine_temperature(°C)'].fillna(test['engine_temperature(°C)'].mean(),inplace=True)
test['motor_torque(N-m)'].fillna(test['motor_torque(N-m)'].mean(),inplace=True)
test['wind_direction(°)'].fillna(test['wind_direction(°)'].mean(),inplace=True)

train.dropna(how='any',axis=0,inplace=True)

train['cloud_level'].replace(['Extremely Low', 'Low', 'Medium'],[0, 1, 2],inplace=True)
test['cloud_level'].replace(['Extremely Low', 'Low', 'Medium'],[0, 1, 2],inplace=True)
train['turbine_status'].value_counts()
test['turbine_status'].value_counts()
dummy = ['turbine_status']
train_dummy = pd.get_dummies(train[dummy])
test_dummy = pd.get_dummies(test[dummy])
train_dummy
test_dummy
train = pd.concat([train,train_dummy],axis=1)
test = pd.concat([test,test_dummy],axis=1)

train["datetime"] = pd.to_datetime(train["datetime"])
test["datetime"] = pd.to_datetime(test["datetime"])

train['dmonth'] = train['datetime'].dt.month
train['dday'] = train['datetime'].dt.day
train['ddayofweek'] = train['datetime'].dt.dayofweek

test['dmonth'] = test['datetime'].dt.month
test['dday'] = test['datetime'].dt.day
test['ddayofweek'] = test['datetime'].dt.dayofweek
```

```
X = train.drop(['tracking_id','datetime','windmill_generated_power(kW/h)','turbine_status'],axis=1)
y = train['windmill_generated_power(kW/h)']

print(X.shape, y.shape)

testData = test.drop(['tracking_id','datetime','turbine_status'],axis=1)
print(testData.shape)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
testData = sc.transform(testData)

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.8,random_state=0)
print(X_train.shape,y_train.shape)
print(X_test.shape,y_test.shape)

from sklearn.tree import DecisionTreeRegressor
regressor_dt = DecisionTreeRegressor(random_state = 42)
regressor_dt.fit(X_train, y_train)

y_train_pred_dt = regressor_dt.predict(X_train)
y_test_pred_dt = regressor_dt.predict(X_test)

print(r2_score(y_true=y_train,y_pred=y_train_pred_dt))
print(r2_score(y_true=y_test,y_pred=y_test_pred_dt))
```

```
from sklearn.ensemble import RandomForestRegressor

regressor_rf = RandomForestRegressor(n_estimators=200, n_jobs=1, oob_score=True,
random_state=42)

regressor_rf.fit(X_train, y_train)


y_train_pred_rf = regressor_rf.predict(X_train)

y_test_pred_rf = regressor_rf.predict(X_test)


print(r2_score(y_true=y_train,y_pred=y_train_pred_rf))

print(r2_score(y_true=y_test,y_pred=y_test_pred_rf))



from xgboost import XGBRegressor

regressor_xg = XGBRegressor(n_estimators=1000, max_depth=8, booster='gbtree', n_jobs=1,
learning_rate=0.1, reg_lambda=0.01, reg_alpha=0.2)

regressor_xg.fit(X_train, y_train)


y_train_pred_xg = regressor_xg.predict(X_train)

y_test_pred_xg = regressor_xg.predict(X_test)


print(r2_score(y_true=y_train,y_pred=y_train_pred_xg))

print(r2_score(y_true=y_test,y_pred=y_test_pred_xg))


model = regressor_xg.predict(testData)

model

model.shape


Ywrite=pd.DataFrame(model,columns=['windmill_generated_power(kW/h)'])

var =pd.DataFrame(test[['tracking_id','datetime']])
```

```
dataset_test_col = pd.concat([var,Ywrite], axis=1)
dataset_test_col.to_csv("static/dataset/Prediction.csv",index=False)
```

8. TESTING

a. Test cases:

	Date	24-Nov-22
	Team ID	PNT2022TMID42592
	Project Name	Predicting the energy output of a wind turbine based on weather condition
	Maximum Marks	4 marks

Test c	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Make sure the user can view the Login/Sign up box after clicking Test	1.Enter the url and page will be directed 2.Make sure you can able to see login/register page		The login page should appear.	Working	Pass	Working	Y		Pasupathi Dadeeja
LoginPage_TC_OO2	Functional	Home Page	Make sure the user can view the	1.Enter username/email in Email text box	Username: admin password: admin	The user should go to the dashboar	Working	Pass	Working	Y		Pasupathi Dadeeja

			Login/Sign up box after clicking Admin	2.Enter password in password text box 3.Enter with valid credentials 4.Click on login button		d.						
LoginPage_TC_OO3	Functional	Login page	Check if the user can access the application with InValid credentials.	1.Enter invalid username/email in Email text box 2.Enter valid password in password text box 3.Click on login button	Username: admin@gmail.con password: admin	Application should show 'Invalid Login credentials'	Working	Pass	Working	Y		Pasupathi Dadeeja
LoginPage_TC_OO4	Functional	Login page	Check if the user can access the application with InValid credentials.	1.Enter valid username/email in Email text box 2.Enter invalid password in password text box 3.Click on login button	Username: admin@gmail.com password: Testing1234576798	Application should show 'Incorrect password'	Working	Pass	Working	Y		Nagaraj G
LoginPage_TC_OO5	Functional	Login page	Check to see if the user can access the programme without an email	1.Leave the Gmail text box as empty 2.Enter password in password text box 3.Click on login button	Username: password: admin	Application should show username is empty	Working	Pass	Working	Y		Nagaraj G

LoginPag	Functional	Login page	Check to see if the user can access the programme without a password	1.Enter username/email in Email text box 2.Leave the password text box as empty 3.Click on Login Button	Username: admin@gmail.com password:	Application should show password is empty	Working	Pass	Working	Y		Nagaraj G
Prediction_Page_001	Functional	Input Page	Make sure the credentials are valid	1.Enter the valid input values 2.Click on Predict Button		The application should show the Predicted value.	Working	Pass	Working	Y		Nagaraj G
Prediction_Page_002	Functional	Input Page	Make sure the credentials are invalid	1.Enter the invalid input values 2.Click on Predict Button		The application should show "Enter valid credentials"	Working	Pass	Working	Y		Palani Selvam M
Logout_Page_001	Functional	Admin logout page	Make sure that the admin has logged out properly	1.Click on logout toggle 2.Log out from the application		The application will show home page again	Working	Pass	Working	Y		Palani Selvam M
Logout_	Functional	User logout page	Make sure that the user has logged out	1.Click on logout toggle 2.Log out from the application		The application will show	Working	Pass	Working	Y		Palani Selvam M

		properly			home page again							

b.User Acceptance Testing:

Purpose of Document

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testingwhere two or more end-users will be involved.

UAT is performed by:

- Client
- End users

Defect Analysis

This reportshows the numberof resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity	Severity	Severity	Severity	Subtotal

	1	2	3	4	
By Design	4	3	2	1	10
Duplicate	1	0	3	0	4
External	2	2	1	1	6
Fixed	4	3	5	19	31
Not Reproduced	1	0	1	1	3
Skipped	0	0	1	1	2
Won't Fix	1	3	2	2	8
Totals	12	11	15	25	64

Test Case Analysis

This report shows the numberof test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Login Page	6	0	0	6
Prediction Page	2	0	0	2
Logout Page	2	0	0	2

9. RESULTS

a. PERFORMANCE METRICS

These metrics are used to track and measure the effectiveness and profitability of various projects. Each stage of the project is tracked and measured againstthe goals that the projectset out to achieve. The data compiledfrom the metricscan be used to plan future projectsand gives insighton how to make projectsmore efficient.

10. ADVANTAGES & DISADVANTAGES

Advantages

- This application is user friendly for all type of users.
- It produces the accurate output wind energy.
- Can store multiple data

Disadvantages

- Difficult to maintain during bad weather.
- Internet connectivity must be present throughout the process.

11.Conclusion

● In this study we showed that wind energy output can be predicted from publicly available weather data with accuracy at best on the training range and at on the unseen test data.

●We demonstrated that an off-the-shelf data modeling and variable selection tool can be used with mostly default settings to run the symbolic regression experiments as well as variable importance.

● The experimental investigations showed the integration of different IBM cloud services. The results show that the responses from the Application were relevant and helpful.

● Although, Web- Application demands complex integration of services it can be deployed easily to leverage the energy prediction and improve the energy production,effectively.

12. Future Scope

● We can include watson studio text to speech and speech to text services to access the applicaton handsfree. This is one of the future scope of this project.

● For future research can be guided to improve relevant results and response time. Simplification of integration of services can be achieved by normalizing the architecture.

●Also, a user feedback system can be incorporated to render the user queries and needs, but still we have given contact us portion for that part.

● There can be storage of responses so that every time it should no search for same answers to already asked questions.

13.APPENDIX

Source Code:

```
# main.py
import os
import base64
import io
import math
from flask import Flask, render_template, Response, redirect, request, session, abort, url_for
import mysql.connector
import hashlib
import datetime
import calendar
import random
from random import randint
from urllib.request import urlopen
import webbrowser

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from werkzeug.utils import secure_filename
from PIL import Image

import urllib.request
import urllib.parse
import socket
import csv

import matplotlib.dates as mdates
import seaborn as sns
from sklearn.metrics import r2_score
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import GradientBoostingRegressor
```

```
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error

import warnings
warnings.filterwarnings('ignore')



mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="",
  charset="utf8",
  database="wind_turbine"

)
app = Flask(__name__)
##session key
app.secret_key = 'abcdef'
#######
UPLOAD_FOLDER = 'static/upload'
ALLOWED_EXTENSIONS = { 'csv'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
#####
@app.route('/', methods=['GET', 'POST'])
def index():
    msg=""


    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM admin WHERE username = %s AND password = %s',
(uname, pwd))
        account = cursor.fetchone()
        if account:
```

```
            session['username'] = uname
            return redirect(url_for('admin'))
        else:
            msg = 'Incorrect username/password!'
    return render_template('index.html',msg=msg)

@app.route('/login', methods=['GET', 'POST'])
def login():
    msg=""


    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM admin WHERE username = %s AND password = %s',
(uname, pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname
            return redirect(url_for('admin'))
        else:
            msg = 'Incorrect username/password!'
    return render_template('login.html',msg=msg)

@app.route('/login_user', methods=['GET', 'POST'])
def login_user():
    msg=""


    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM wt_register WHERE uname = %s AND pass = %s',
(uname, pwd))
```

```
        account = cursor.fetchone()
        if account:
            session['username'] = uname
            return redirect(url_for('test_data'))
        else:
            msg = 'Incorrect username/password!'
    return render_template('login_user.html',msg=msg)

@app.route('/register', methods=['GET', 'POST'])
def register():
    msg=""
    act=request.args.get("act")
    if request.method=='POST':
        name=request.form['name']
        location=request.form['location']
        mobile=request.form['mobile']
        email=request.form['email']
        uname=request.form['uname']
        pass1=request.form['pass']



        mycursor = mydb.cursor()

        now = datetime.datetime.now()
        rdate=now.strftime("%d-%m-%Y")

        mycursor.execute("SELECT count(*) from wt_register where uname=%s",(uname,))
        cnt = mycursor.fetchone()[0]

        if cnt==0:
            mycursor.execute("SELECT max(id)+1 FROM wt_register")
            maxid = mycursor.fetchone()[0]
            if maxid is None:
                maxid=1
```

```
            sql = "INSERT INTO wt_register(id,name,location,mobile,email,uname,pass,create_date)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
            val = (maxid,name,location,mobile,email,uname,pass1,rdate)
            mycursor.execute(sql, val)
            mydb.commit()
            #print(mycursor.rowcount, "Registered Success")
            msg="sucess"
            #if mycursor.rowcount==1:
            return redirect(url_for('register',act='1'))
        else:
            msg='User Already Exist!'
    return render_template('register.html',msg=msg,act=act)

@app.route('/admin', methods=['GET', 'POST'])
def admin():
    msg=""


    pd.pandas.set_option('display.max_columns', None)
    dataset = pd.read_csv('static/dataset/train.csv')
    dat=dataset.head()
    data=[]
    for ss in dat.values:
        data.append(ss)


    return render_template('admin.html',msg=msg,data=data)

@app.route('/load_data', methods=['GET', 'POST'])
def load_data():
    msg=""

    pd.pandas.set_option('display.max_columns', None)
    dataset = pd.read_csv('static/dataset/train.csv')
    dat=dataset.head(200)
```

```
    data=[]
    for ss in dat.values:
        data.append(ss)


    return render_template('load_data.html',data=data)


@app.route('/preprocess', methods=['GET', 'POST'])
def preprocess():
    msg=""

    pd.pandas.set_option('display.max_columns', None)
    dataset = pd.read_csv('static/dataset/train.csv')
    dat=dataset.head(200)
    rows=len(dataset.values)
    data3=[]
    for ss3 in dat.values:
        cnt=len(ss3)

        data3.append(ss3)
    cols=cnt
    mem=float(rows)*0.75

    ##
    list_of_column_names=[]
    with open("static/dataset/train.csv") as csv_file:
        csv_reader = csv.reader(csv_file, delimiter = ',')
        list_of_column_names = []
        for row in csv_reader:
            list_of_column_names.append(row)
            break
    ##

    print(list_of_column_names)
```

```
    dat4=dataset.isna().sum()
    dr=np.stack(dat4)
    #print(dr)

    data4=[]
    i=0
    k=len(dr)
    j=k-1
    for ss4 in dr:
        if i<j:
            dt=[]
            dt.append(list_of_column_names[0][i])
            dt.append(ss4)
            data4.append(dt)
        i+=1

    return
render_template('preprocess.html',mem=mem,rows=rows,cols=cols,data3=data3,data4=data4)



@app.route('/data_analysis', methods=['GET', 'POST'])
def data_analysis():
    msg=""

    pd.pandas.set_option('display.max_columns', None)
    dataset = pd.read_csv('static/dataset/train.csv')
    dat=dataset.head(200)

    null_col = [i for i in dataset.columns if dataset[i].isnull().sum()>0 and dataset[i].dtypes != 'O']
    null_col

    for i in null_col:
        dataset[i].fillna(dataset[i].mean(), inplace=True)
    dataset.isnull().sum()
```

```
dataset.cloud_level.value_counts()
dataset['cloud_level'].fillna('Low', inplace=True)
dataset.isnull().sum()

dataset['turbine_status'].unique()
dataset['turbine_status'].fillna('Missing', inplace=True)
l = ['BA', 'A2', 'ABC', 'AAA', 'BD', 'AC', 'BB', 'BCB', 'B', 'AB', 'Missing', 'B2', 'BBB', 'A', 'D']
feat_tur = dict()
for i in range(len(l)):
    feat_tur[l[i]] = i
feat_tur
dataset['turbine_status']=dataset['turbine_status'].map(feat_tur)

dataset.isnull().sum()
dataset['datetime'] = pd.to_datetime(dataset['datetime'])
dataset['day'] = dataset['datetime'].dt.date
dataset['time'] = dataset['datetime'].dt.time
# dataset.drop('datetime', axis=1, inplace=True)
dataset['day'] = pd.to_datetime(dataset['day'])
dataset['time']= pd.to_datetime(dataset['time'].astype(str))
dataset.dtypes
dataset['date']=dataset['day'].dt.day
dataset['month']=dataset['day'].dt.month
dataset['year']=dataset['day'].dt.year
dataset.drop('day', axis=1, inplace=True)
dataset.head(2)

dataset['time_hour'] = dataset['time'].dt.hour
dataset['time_minute'] = dataset['time'].dt.minute
dataset.drop('time', axis=1, inplace=True)
dataset.head(2)
dataset.dtypes

#
plt_feat = ['datetime', 'wind_speed(m/s)',
  'atmospheric_temperature(°C)', 'shaft_temperature(°C)',
```

```
        'blades_angle(°)', 'gearbox_temperature(°C)', 'engine_temperature(°C)',
        'motor_torque(N-m)', 'generator_temperature(°C)',
        'atmospheric_pressure(Pascal)', 'area_temperature(°C)',
        'windmill_body_temperature(°C)', 'wind_direction(°)', 'resistance(ohm)',
        'rotor_torque(N-m)', 'cloud_level', 'blade_length(m)',
        'blade_breadth(m)', 'windmill_height(m)']

    '''import matplotlib.pyplot as plt
    import seaborn as sns
    for i in plt_feat:
        plt.figure(figsize=(8,4))
        sns.scatterplot(data=dataset, x=i, y='windmill_generated_power(kW/h)')
        plt.show()'''


    return render_template('data_analysis.html')



@app.route('/feature_extract', methods=['GET', 'POST'])
def feature_extract():
    msg=""

    df_train = pd.read_csv(r"static/dataset/train.csv")
    df_test= pd.read_csv(r"static/dataset/test.csv")
    df_train.head()
    df_train.nunique()
    df_test.nunique()
    df_train.isna().sum()
    df_test.isna().sum()
    dat=df_train.corr()
    #print(dat)

col=['wind_speed(m/s)','atmospheric_temperature(C)','shaft_temperature(C)','blades_angle','gear
box_temperature(C)','engine_temperature(C)','motor_torque(N-
m)','generator_temperature(C)','atmospheric_pressure(Pascal)','area_temperature(C)','windmill_b
```

```
ody_temperature(C)','wind_direction','resistance(ohm)','rotor_torque(N-
m)','blade_length(m)','blade_breadth(m)','windmill_height(m)']
    data=[]
    #col=list(df_train.columns)
    i=0
    for ss3 in dat.values:
        dt=[]
        if i<17:
            dt.append(col[i])
            dt.append(ss3)
            data.append(dt)
        i+=1

    #
    corr = df_train.corr()
    corr.style.background_gradient(cmap='coolwarm')
    def splitFeatures(df):
        numerical_features = df.select_dtypes(include=[np.number])
        categorical_features = df.select_dtypes(include=[np.object])
        return numerical_features, categorical_features

    numerical_features,categorical_features=splitFeatures(df_train)
    numerical_features
    dat2=categorical_features
    j=0
    data2=[]
    for ss2 in categorical_features.values:
        if i<200:
            data2.append(ss2)
        j+=1

    ####
    df_cpy = df_train.copy()
    def comparing_train_and_test_feature(df,df_test,col):
        fig = plt.figure(figsize=(16,10))
        ax0 = fig.add_subplot(1,2,1)
```

```
        ax1 = fig.add_subplot(1,2,2)
        df[col].plot(kind='kde',ax=ax0)
        df_test[col].plot(kind='kde',ax=ax1)
        ax0.set_xlabel(col)
        ax1.set_xlabel(col)
        ax0.set_title("Density plot of " + str(col) + " of training set")
        ax1.set_title("Density plot of " + str(col) + " of testing set")
        #plt.show()

    #comparing_train_and_test_feature(df_train,df_test,'wind_speed(m/s)')

#sns.scatterplot(x='wind_speed(m/s)',y='windmill_generated_power(kW/h)',hue='cloud_level',data=df_train)
    #plt.show()

    #comparing_train_and_test_feature(df_train,df_test,'atmospheric_temperature(°C)')

#sns.scatterplot(x='atmospheric_temperature(°C)',y='windmill_generated_power(kW/h)',hue='cloud_level',data=df_train)
    #plt.show()
    #plt.close()
    #comparing_train_and_test_feature(df_train,df_test,'shaft_temperature(°C)')

'''sns.scatterplot(x='shaft_temperature(°C)',y='windmill_generated_power(kW/h)',hue='cloud_level',data=df_train)
    plt.show()
    plt.close()
    #comparing_train_and_test_feature(df_train,df_test,'blades_angle(°)')

sns.scatterplot(x='gearbox_temperature(°C)',y='windmill_generated_power(kW/h)',hue='cloud_level',data=df_train)
    plt.show()
    plt.close()'''
```

```
    return render_template('feature_extract.html',data=data,data2=data2)


@app.route('/classify', methods=['GET', 'POST'])
def classify():
    msg=""

    train = pd.read_csv('static/dataset/train.csv')
    test = pd.read_csv('static/dataset/test.csv')
    train.head()
    test.head()
    train.describe()

    test.describe()
    ##
    corr = train.corr()
    plt.figure(figsize=(20,10))
    mask = np.zeros_like(corr,dtype=np.bool)
    mask[np.triu_indices_from(mask)] = True
    #sns.heatmap(corr,mask=mask,annot=True)
    #plt.show()

    ###

train.drop(['generator_temperature(°C)','windmill_body_temperature(°C)'],inplace=True,axis=1)

test.drop(['generator_temperature(°C)','windmill_body_temperature(°C)'],inplace=True,axis=1)

    train.isnull().sum()
    test.isnull().sum()
    #sns.heatmap(train.isnull(),cbar=False,yticklabels=False,cmap = 'viridis')
    #sns.heatmap(test.isnull(),cbar=False,yticklabels=False,cmap = 'viridis')



    train['gearbox_temperature(°C)'].fillna(train['gearbox_temperature(°C)'].mean(),inplace=True)
```

```
    train['area_temperature(°C)'].fillna(train['area_temperature(°C)'].mean(),inplace=True)
    train['rotor_torque(N-m)'].fillna(train['rotor_torque(N-m)'].mean(),inplace=True)
    train['blade_length(m)'].fillna(train['blade_length(m)'].mean(),inplace=True)
    train['blade_breadth(m)'].fillna(train['blade_breadth(m)'].mean(),inplace=True)
    train['windmill_height(m)'].fillna(train['windmill_height(m)'].mean(),inplace=True)
    train['cloud_level'].fillna(train['cloud_level'].mode()[0],inplace=True)

train['atmospheric_temperature(°C)'].fillna(train['atmospheric_temperature(°C)'].mean(),inplace=
True)

train['atmospheric_pressure(Pascal)'].fillna(train['atmospheric_pressure(Pascal)'].mean(),inplace
=True)
    train['wind_speed(m/s)'].fillna(train['wind_speed(m/s)'].mean(),inplace=True)
    train['shaft_temperature(°C)'].fillna(train['shaft_temperature(°C)'].mean(),inplace=True)
    train['blades_angle(°)'].fillna(train['blades_angle(°)'].mean(),inplace=True)
    train['engine_temperature(°C)'].fillna(train['engine_temperature(°C)'].mean(),inplace=True)
    train['motor_torque(N-m)'].fillna(train['motor_torque(N-m)'].mean(),inplace=True)
    train['wind_direction(°)'].fillna(train['wind_direction(°)'].mean(),inplace=True)

    test['gearbox_temperature(°C)'].fillna(test['gearbox_temperature(°C)'].mean(),inplace=True)
    test['area_temperature(°C)'].fillna(test['area_temperature(°C)'].mean(),inplace=True)
    test['rotor_torque(N-m)'].fillna(test['rotor_torque(N-m)'].mean(),inplace=True)
    test['blade_length(m)'].fillna(test['blade_length(m)'].mean(),inplace=True)
    test['blade_breadth(m)'].fillna(test['blade_breadth(m)'].mean(),inplace=True)
    test['windmill_height(m)'].fillna(test['windmill_height(m)'].mean(),inplace=True)
    test['cloud_level'].fillna(test['cloud_level'].mode()[0],inplace=True)

test['atmospheric_temperature(°C)'].fillna(test['atmospheric_temperature(°C)'].mean(),inplace=T
rue)

test['atmospheric_pressure(Pascal)'].fillna(test['atmospheric_pressure(Pascal)'].mean(),inplace=T
rue)
    test['wind_speed(m/s)'].fillna(test['wind_speed(m/s)'].mean(),inplace=True)
    test['shaft_temperature(°C)'].fillna(test['shaft_temperature(°C)'].mean(),inplace=True)
    test['blades_angle(°)'].fillna(test['blades_angle(°)'].mean(),inplace=True)
    test['engine_temperature(°C)'].fillna(test['engine_temperature(°C)'].mean(),inplace=True)
```

```
    test['motor_torque(N-m)'].fillna(test['motor_torque(N-m)'].mean(),inplace=True)
    test['wind_direction(°)'].fillna(test['wind_direction(°)'].mean(),inplace=True)

    train.dropna(how='any',axis=0,inplace=True)

    train['cloud_level'].replace(['Extremely Low', 'Low', 'Medium'],[0, 1, 2],inplace=True)
    test['cloud_level'].replace(['Extremely Low', 'Low', 'Medium'],[0, 1, 2],inplace=True)
    train['turbine_status'].value_counts()
    test['turbine_status'].value_counts()
    dummy = ['turbine_status']
    train_dummy = pd.get_dummies(train[dummy])
    test_dummy = pd.get_dummies(test[dummy])
    train_dummy
    test_dummy
    train = pd.concat([train,train_dummy],axis=1)
    test = pd.concat([test,test_dummy],axis=1)

    train["datetime"] = pd.to_datetime(train["datetime"])
    test["datetime"] = pd.to_datetime(test["datetime"])

    train['dmonth'] = train['datetime'].dt.month
    train['dday'] = train['datetime'].dt.day
    train['ddayofweek'] = train['datetime'].dt.dayofweek

    test['dmonth'] = test['datetime'].dt.month
    test['dday'] = test['datetime'].dt.day
    test['ddayofweek'] = test['datetime'].dt.dayofweek

    ##Auto Regression
    '''X =
train.drop(['tracking_id','datetime','windmill_generated_power(kW/h)','turbine_status'],axis=1)
    y = train['windmill_generated_power(kW/h)']

    print(X.shape, y.shape)

    testData = test.drop(['tracking_id','datetime','turbine_status'],axis=1)
```

```
    print(testData.shape)
    from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    X = sc.fit_transform(X)
    testData = sc.transform(testData)

    from sklearn.model_selection import train_test_split
    X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.8,random_state=0)
    print(X_train.shape,y_train.shape)
    print(X_test.shape,y_test.shape)

    from sklearn.tree import DecisionTreeRegressor
    regressor_dt = DecisionTreeRegressor(random_state = 42)
    regressor_dt.fit(X_train, y_train)

    y_train_pred_dt = regressor_dt.predict(X_train)
    y_test_pred_dt = regressor_dt.predict(X_test)

    print(r2_score(y_true=y_train,y_pred=y_train_pred_dt))
    print(r2_score(y_true=y_test,y_pred=y_test_pred_dt))

    from sklearn.ensemble import RandomForestRegressor
    regressor_rf = RandomForestRegressor(n_estimators=200, n_jobs=1, oob_score=True,
random_state=42)
    regressor_rf.fit(X_train, y_train)

    y_train_pred_rf = regressor_rf.predict(X_train)
    y_test_pred_rf = regressor_rf.predict(X_test)

    #print(r2_score(y_true=y_train,y_pred=y_train_pred_rf))
    #print(r2_score(y_true=y_test,y_pred=y_test_pred_rf))



    from xgboost import XGBRegressor
    regressor_xg = XGBRegressor(n_estimators=1000, max_depth=8, booster='gbtree', n_jobs=1,
learning_rate=0.1, reg_lambda=0.01, reg_alpha=0.2)
```

```
    regressor_xg.fit(X_train, y_train)

    y_train_pred_xg = regressor_xg.predict(X_train)
    y_test_pred_xg = regressor_xg.predict(X_test)


    print(r2_score(y_true=y_train,y_pred=y_train_pred_xg))
    print(r2_score(y_true=y_test,y_pred=y_test_pred_xg))"'

    "'model = regressor_xg.predict(testData)
    model
    model.shape

    Ywrite=pd.DataFrame(model,columns=['windmill_generated_power(kW/h)'])
    var =pd.DataFrame(test[['tracking_id','datetime']])
    dataset_test_col = pd.concat([var,Ywrite], axis=1)
    dataset_test_col.to_csv("static/dataset/Prediction.csv",index=False)"'
    ff=open("static/det.txt","r")
    v=ff.read()
    vv=v.split(',')
    dataset = pd.read_csv('static/dataset/Prediction.csv')
    dat=dataset.head(200)

    data=[]
    for ss in dat.values:
        data.append(ss)

    return render_template('classify.html',vv=vv,data=data)


@app.route('/test_data', methods=['GET', 'POST'])
def test_data():
    act=""
    res=""
    uname=""
    if 'username' in session:
```

```
    uname = session['username']

if uname is None:
    return redirect(url_for('login'))

mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM wt_register where uname=%s",(uname,))
data = mycursor.fetchone()
name=data[1]

ws=""
atmos_temp=""
gear_temp=""
eng_temp=""
gen_temp=""
atmos_pres=""
area_temp=""
wind_dir=""
turbine_st=""
cloud=""

if request.method=='POST':
    ws=request.form['ws']
    atmos_temp=request.form['atmos_temp']
    gear_temp=request.form['gear_temp']
    eng_temp=request.form['eng_temp']
    gen_temp=request.form['gen_temp']
    atmos_pres=request.form['atmos_pres']
    area_temp=request.form['area_temp']
    wind_dir=request.form['wind_dir']
    turbine_st=request.form['turbine_st']
    cloud=request.form['cloud']


    df = pd.read_csv("static/dataset/train.csv")
```

```
x=0

ws1=float(ws)
w1=ws1-3
w2=ws1+3

at1=float(atmos_temp)
a1=at1-3
a2=at1+3

gr=float(gear_temp)
gr1=gr-3
gr2=gr+3

eg=float(eng_temp)
eg1=eg-3
eg2=eg+3

gn=float(gen_temp)
gn1=gn-3
gn2=gn+3

ap=float(atmos_pres)
ap1=ap-3
ap2=ap+3

at=float(area_temp)
at1=at-3
at2=at+3

wd=float(wind_dir)
wd1=wd-3
wd2=wd+3
```

```
        for rr in df.values:
            if rr[2]>w1 and rr[2]<=w2 and rr[3]>a1 and rr[3]<=a2 and rr[6]>gr1 and rr[6]<=gr2 and
rr[7]>eg1 and rr[7]<=eg2:
                print("a")
                if rr[9]>gn1 and rr[9]<gn2 and rr[10]>ap1 and rr[10]<=ap2 and rr[11]>at1 and
rr[11]<=at2 and rr[13]>wd1 and rr[13]<=wd2:
                    print("b")
                    if rr[16]==turbine_st and rr[17]==cloud:
                        print("c")
                        res=rr[21]
                        x+=1
                        break

        print(x)

        if x>0:
            act="1"
        else:
            act="2"


    return render_template('test_data.html',name=name,res=res,act=act)




@app.route('/logout')
def logout():
    # remove the username from the session if it is there
    session.pop('username', None)
    return redirect(url_for('index'))



if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

Git Repo Link:

https://github.com/IBM-EPBL/IBM-Project-21426-1659780053

Demo Link:

click here to watch the demo video:

https://youtu.be/dmCVw-voXtU