HINDUSTHAN INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai, Accredited with "A" Grade by NAAC) Valley Campus, Pollachi Main Road, Coimbatore 641 032.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

REPORT ON

HX 8001 PROFESSIONAL READINESS FOR INNOVATION,

EMPLOYABILITY AND ENTREPRENEURSHIP

(Naalaiya Thiran Program)

PROJECT TITLE

PREDICTING THE ENERGY OUTPUT OF WIND TURBINE BASED ON WEATHER CONDITION

TEAM ID: PNT2022TMID10490

TEAM MEMBERS

1.D.HEMANATHAN(TEAM LEADER)

2.GURRAM KUSUMA

3.D.HEMAMALINI

4.S.HEMANTH

MENTOR

Mrs.S.BRINDHA

EVALUATOR

Dr.S.CHINNAPPARAJ

1. INTRODUCTION

1.1 PROJECT REVIEW

Wind speed/power has received increasing attentionaround the earth due to its renewable nature as well asenvironmental friendliness. With the global installed windpower capacity rapidly increasing, the wind industry is growing into a large-scale business. Reliable short-term wind speed forecasts play a practical and crucial role in wind energy conversion systems, such as the dynamic control of wind turbines and power system scheduling. A precise forecast needs to overcome problems of variable energy production caused by fluctuating weather conditions. Power generated by wind is highly dependent on the wind speed. Though it is highly non-linear, wind speed follows a certain pattern over a certain period of time.We exploit this time series pattern to gain useful information and use it for power prediction.

1.2 PURPOSE

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

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

S.NO	TITLE	AUTHOR	YEAR	PROPOSED SYSTEM
1	A Multi-Step Prediction Method for Wind Power Based on Improved TCN to Correct Cumulative Error	Heifeng luo, Xun dou, Rongsun	2021	A Multi-step wind power prediction method was proposed by exploiting improved TCN to correct the cumulative error. First, multi-scale convolution (MSC) and self-attentiveness (SA) were adopted to optimize the problem that a single-scale convolution kernel of TCN is difficult to extract temporal and spatial features at different scales of the input sequence.
2	Remotely Sensed Winds and Wind Stresses for Marine Forecasting and Ocean Modeling	Mark A. Bourassa ,Thomas Meissner ,Ivana Cerovecki	2019	Remotely sensed surface winds (scalar winds and vector winds) with related material on surface stress, air-sea heat fluxes, currents, sea state, and precipitation.
3	Wind Generation Forecasting Methods and Proliferation of Artificial Neural Network	MuhammadShahzad Nazir, Fahad Alturise, Sami Alshmrany	2020	Wind forecasting methods and the artificial neural network, The instrument used to measure wind assimilation is analyzed and discussed, accurately. The high forecasting accuracy could be achieved through proper handling and calibration of the wind-

2.2 REFERENCES

1.Luo H, Dou X, Sun R and Wu S (2021) A Multi-Step Prediction Method for Wind Power Based on Improved TCN to Correct Cumulative Error. *Front. Energy Res.* 9:723319. doi: 10.3389/fenrg.2021.723319.

2. Bourassa MA, Meissner T, Cerovecki I, Chang PS, Dong X, De Chiara G, Donlon C, Dukhovskoy DS, Elya J, Fore A, Fewings MR, Foster RC, Gille ST, Haus BK, Hristova-Veleva S, Holbach HM, Jelenak Z, Knaff JA, Kranz SA, Manaster A, Mazloff M, Mears C, Mouche A, Portabella M, Reul N, Ricciardulli L, Rodriguez E, Sampson C, Solis D, Stoffelen A, Stukel MR, Stiles B, Weissman D and Wentz F (2019) Remotely Sensed Winds and Wind Stresses for Marine Forecasting and Ocean Modeling. *Front. Mar. Sci.* 6:443. doi: 10.3389/fmars.2019.00443.

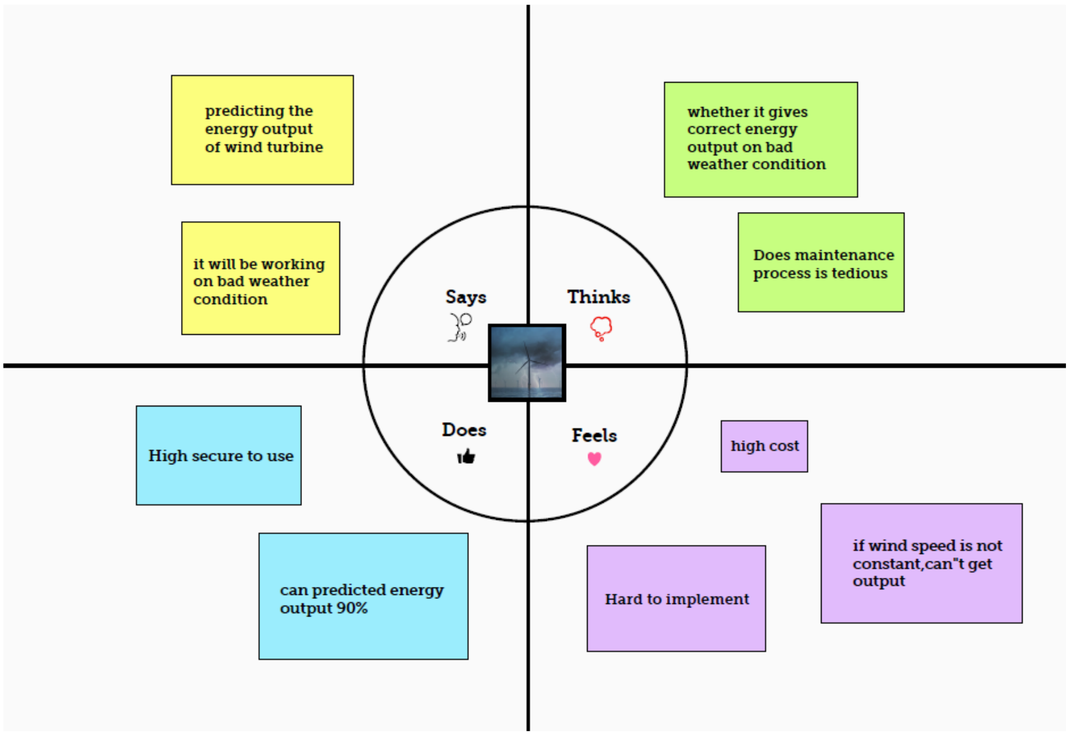
3.Nazir, Muhammad Shahzad, Fahad Alturise, Sami Alshmrany, Hafiz. M. J Nazir, Muhammad Bilal, Ahmad N. Abdalla, P. Sanjeevikumar, and Ziad M. Ali. 2020. "Wind Generation Forecasting Methods and Proliferation of Artificial Neural Network: A Review of Five Years Research Trend" *Sustainability* 12, no. 9: 3778. https://doi.org/10.3390/su12093778.

2.3 PROBLEM STATEMENT DEFINITION

The prediction of wind power plays an indispensable role in maintaining the stability of the entire power grid. Due to its renewable resources and environmental friendliness, wind speed/power has gained increasing interest worldwide. The wind industry is rapidly expanding into a large-scale industry as a result of the fast-rising amount of installed wind generating capacity worldwide. When it comes to scheduling power systems and other practical aspects of wind energy conversion, such as the dynamic management of wind turbines, reliable short- term wind speed forecasts are essential. A precise forecast is required to solve issues with variable energy production brought on by changing weather patterns. The wind speed has a big impact on how much power is produced by the wind. Despite being quite nonlinear, wind speed exhibits a consistent pattern over a specific amount of time. 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.

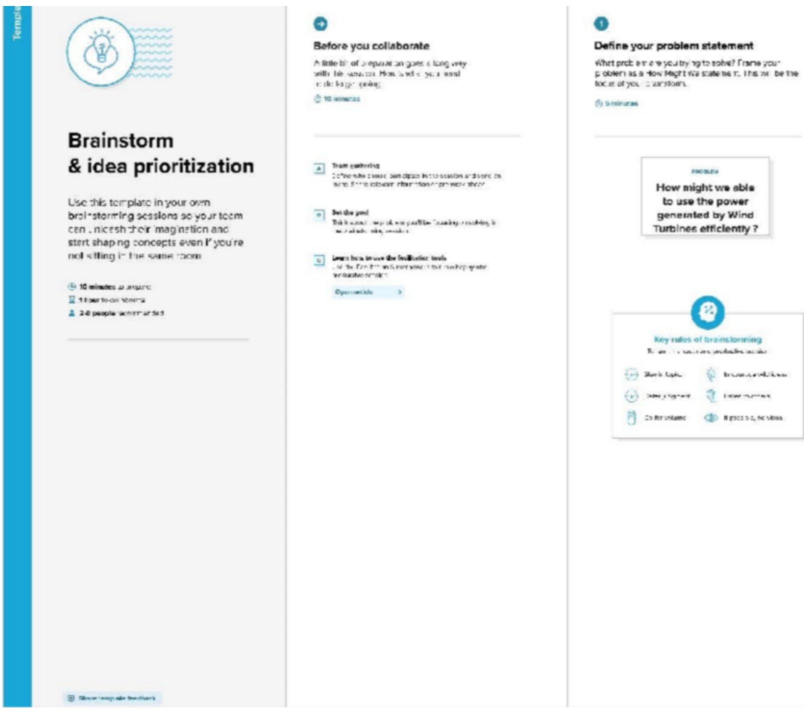
3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

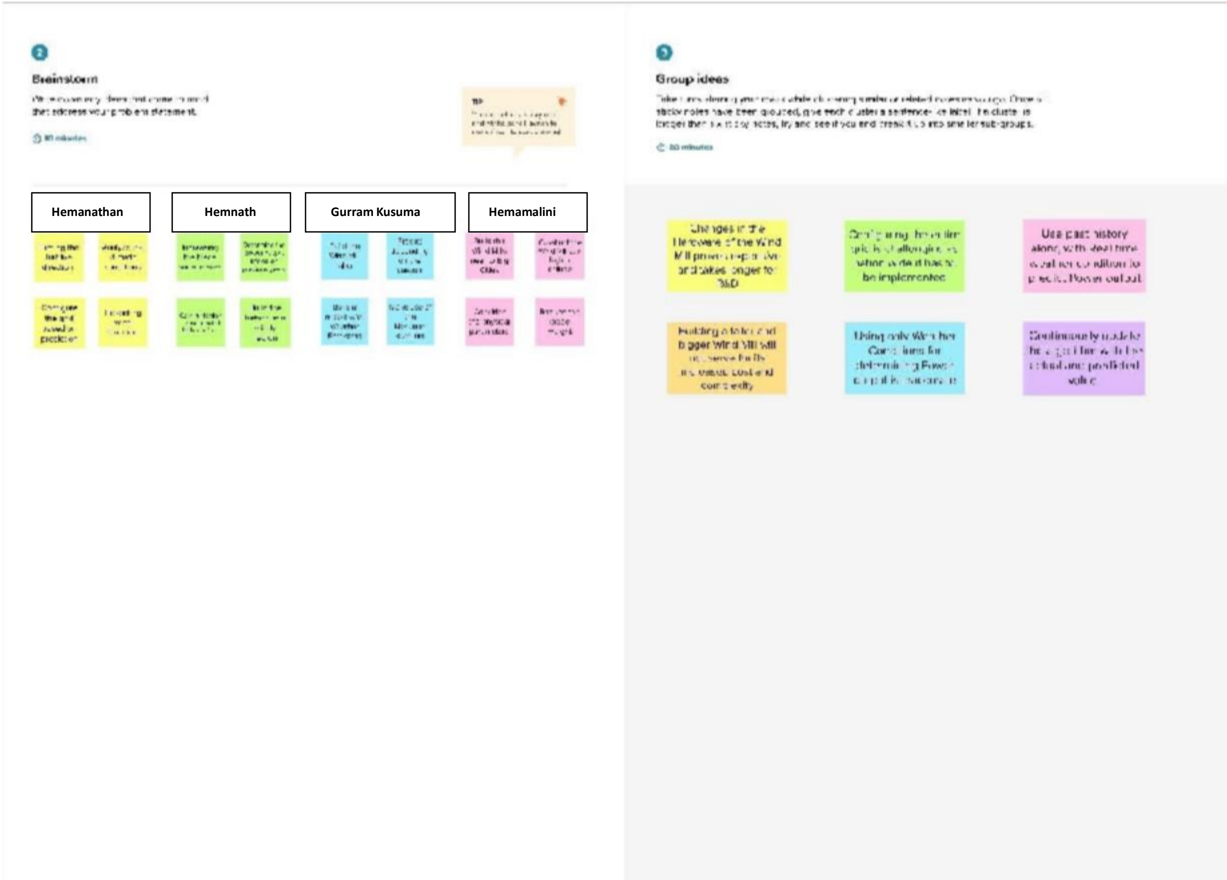


3.2 IDEATION AND BRAINSTROMING

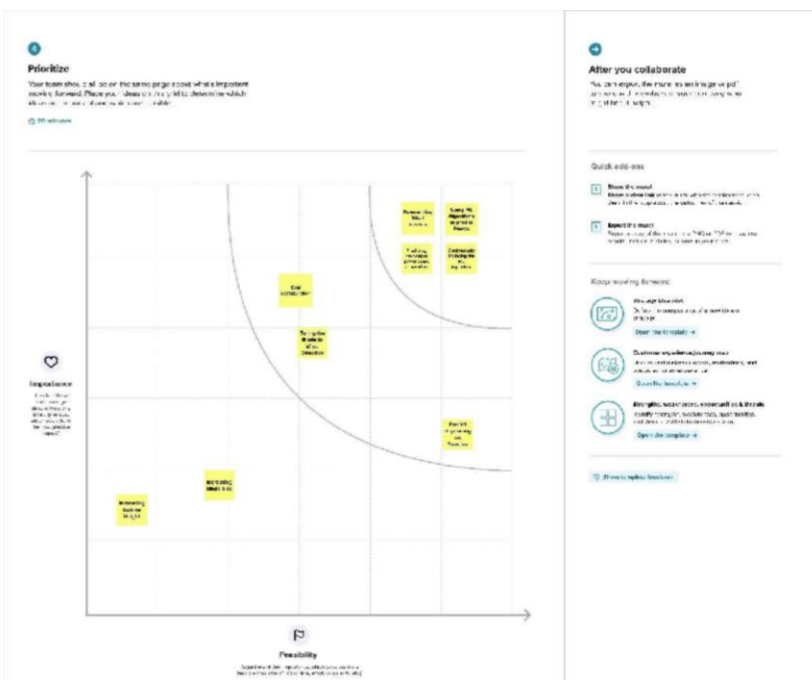
STEP 1:



Step 2: Brainstorm, Idea listing and Groupin

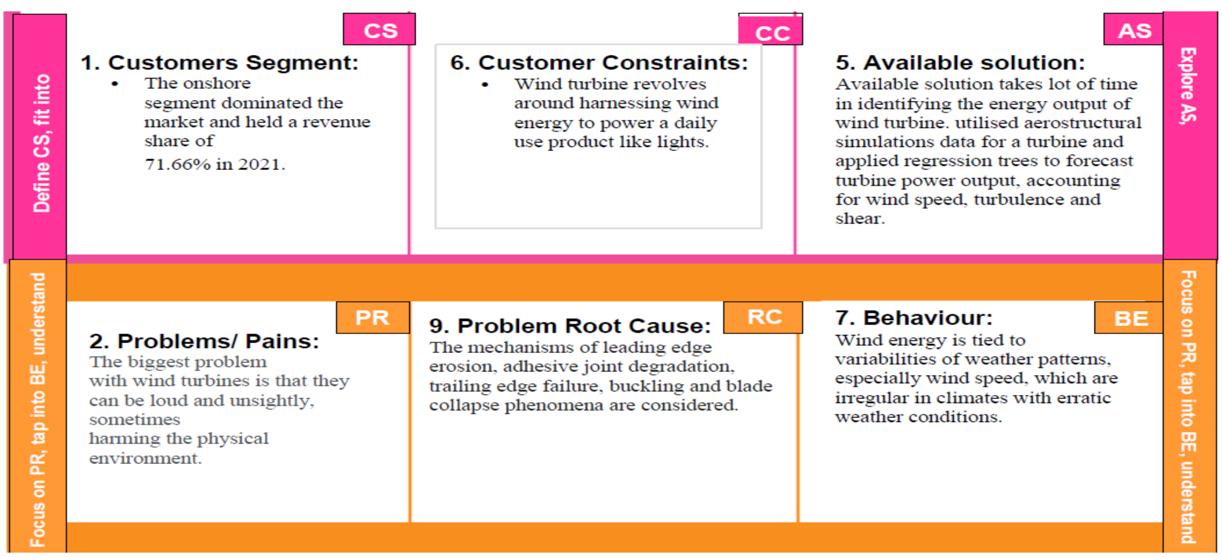


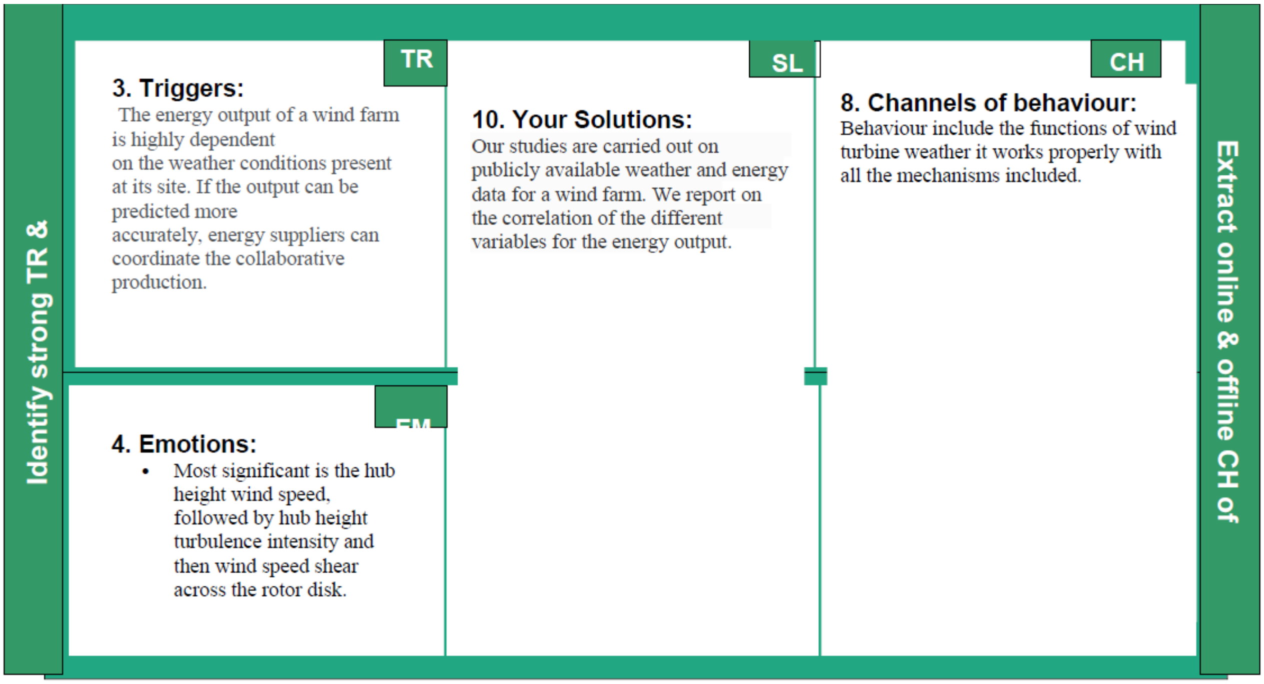
Step 3: Idea Prioritization



S.No.	Parameter	Description
1.	Problem Statement (Problemto be solved)	Our aim is to map weather data to energy production. The model prediction is then showcased on user interface to predict the energy outputof wind turbine.
2.	Idea / Solution description	Our approach was to use a time seriesforecasting model that would generate point forecast of wind generation for the upcoming three days, for a wind turbine.
3.	Novelty / Uniqueness	It will be working on bad weathercondition. Precise information on timing Flectuation in weatherconditions
4.	Social Impact / CustomerSatisfaction	Wind energy jobs in rural communitiesin manufacturing, transportation and project construction.

3.4 PROBLEM SOLUTION FIT





4. REQUIERMENT ANALYSIS

4.1 FUNCTIONAL REQUIERMENT

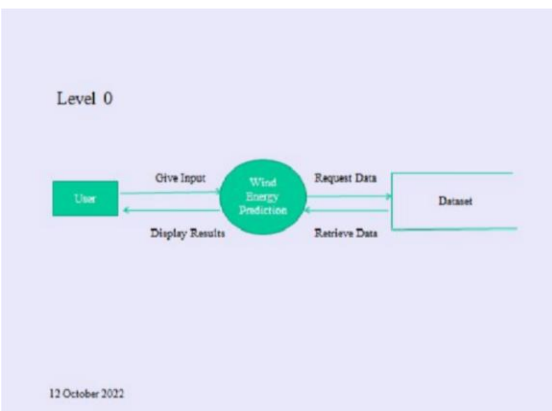
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration and logging in by entering their username and password.	Registration through Form.
FR-2	User Confirmation by validatingthe username with respect to the password	Confirmation via pop-up Message.
FR-3	Displaying the further information about the application.	By selecting the about button the details of theapplication will be displayed.
FR-4	Validating the city name.	System checks whether the city entered by the user ispresent or not. If present it will collect the further details else it will display the pop-up message as errorin the city.
FR-5	Checking the data type of the value.	System checks for the data type of the value entered bythe user.
FR-6	Validating all required fields.	Before predicting the output the system checks whether all the values are entered by the user and checks whether all values are correct.

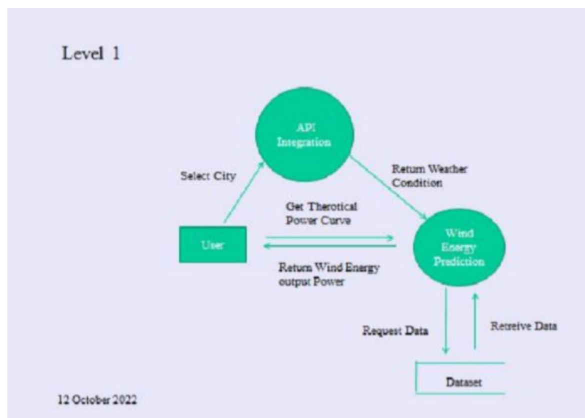
4.2 NON FUNCTIONAL REQUIERMENTS

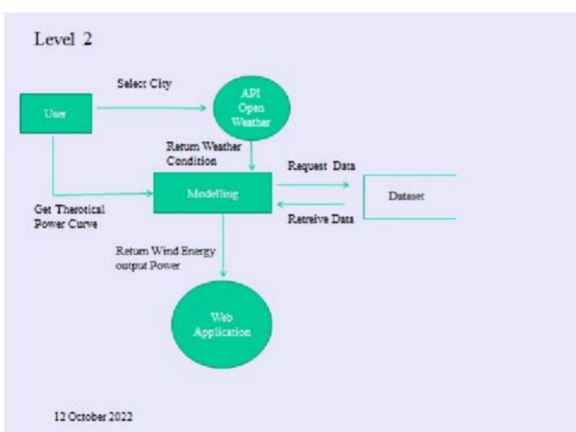
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system satisfies the user goals and the application is easy to use.
NFR-2	**Security**	The data provided to system will be protected fromattacks and unauthorized access
NFR-3	**Reliability**	The system will provide the consistency in output without producing an error.
NFR-4	**Performance**	The performance will never degrade even the workload is increased.
NFR-5	**Availability**	The application is available for 24*7
NFR-6	**Scalability**	The system can be used as web application as well as mobile application with a sufficient internet availability.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS







5.2 SOLUTION & TECHNICAL ARCHITECTURE

The proposed communication network architecture for the Smart-WPF consists of three networks: the turbine area network (TAN), the farm area network (FAN), and the control area network (CAN). It consists of hierarchical architectures where Level 1 is a sensor network in a single wind turbine, Level 2 is the wind turbine- to-wind turbine interaction in the WPF, Level 3 isthe local control center to wind turbine interaction, and Level 4 is the farm- to-farm interaction to optimize grid operation. In order to implement hierarchical network architectures, a hybrid communication solution is considered. EPON-based architecture represents a wired solution, while ZigBee-Pro is considered for the wireless solution. In this work, Levels 1 and 2 are explained in more detail, while Levels 3 and 4 are out the scope of this work.

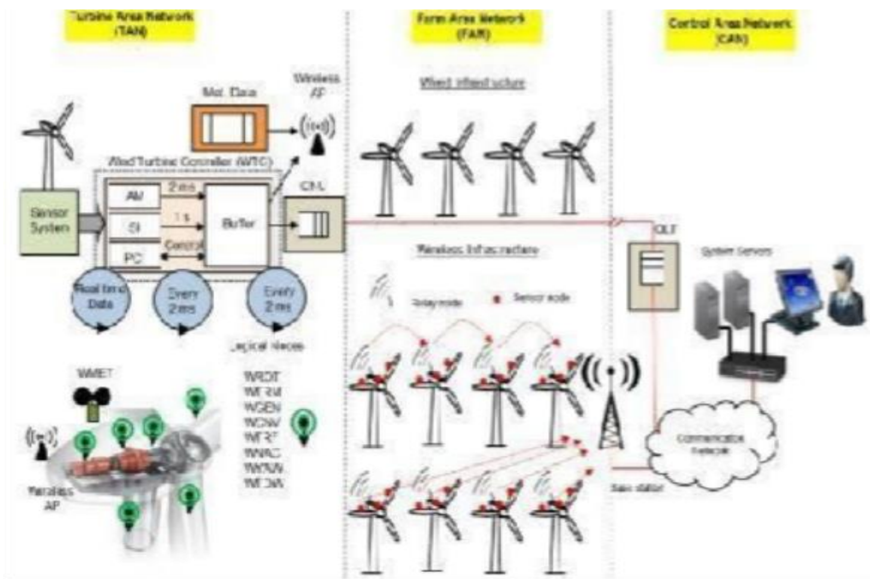


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	This is used by the user for interacting with the system to know about the services provided by system.	HTML, CSS, Angular Js.
2.	Weather Data collector	This weather data collector is used to collect thereal time weather data in the environment.	Sensors, wired and wireless network.
3.	Symbolic Regression (MachineLearning Model)	To deal with interaction of the different parameters.	Genetic Programming Data Modeler.

4.	Database	Used to store the collected and examine weathercondition and energy output.	MySQL, NoSQL, etc.
5.	File Storage	To store the data files in the databases for futurereferences.	Local or Global File System or IBMStorage.
6.	External API	This application programming interfaces is used toknow about the energy output based on every weather condition.	Weather conditions obtained and theirenergy output.
7.	Infrastructure (Server / Cloud)	The whole system is applied and stored in serverfor easy access and retrieved.	Data Storage Server or IBM CloudServers

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The open source framework used in this system isflexible and it includes R, python etc..	IBM Open source Tools and databases.
2.	Security Implementations	The data stored in the database when shared withindustries are encrypted and shared as encrypteddata to avoid the access of data by third party people.	SHA-256, Encryptions, IAMControls, OWASP etc.
3.	Scalable Architecture	The architecture used here is a 3tier architecture where a middleware is presentto carry out the communication between client and server.	3tier architecture.
4.	Availability	The system is designed in a way that It can handle traffic in a better way and thus helps the system available for users at any time.	Network traffic analysis tools.
5.	Performance	The system can efficiently handle a higher number of request and can also uses catch buffer to store and retrieve the data in a easier way.	Methods like Confusion Matrix F1 score, Precision – Recall curve etc.

5.3 USER STORIES

User Type	Functional Requirements	User Number Story	User Story/User Task	Acceptance Criteria	Priority	Release
Customer	Home(Application)	USN-1	As a user, I can view the guideline as well as the detailed information about the application	I can gain knowledge by practical method to use this application .	Low	Sprint-1
		USN-2	As a User, I can use this application by reading the instructions	I can use this in user friendly method by reading the instruction.	Low	Sprint-1
		USN-3	As a User, I can login and by entering the correct username and password	If login is correctly entered ,I can navigate to the next page.	Low	Sprint-2
		USN-4	As a user ,I am allowed to select the city and can get the weather of the city.	I can select the city ,If the city is correct I can further enter the details.	Medium	Sprint-3
		USN-5	As a user I am allowed to view the weather of the selected city.	If correct city is selected ,then the weather of the particular city will be displayed.	Medium	Sprint-4
		USN-6	As a User ,I can view the Power generated by the wind	If all values are entered correctly I can view the power generated by the	High	Sprint-5
		USN-7	As a User, I can use the web application virtually anywhere	I can use the application portably	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & DELIVERY

Sprint	Functional Requirement(EPIC)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user ,I canregister for the application by entering email, password and conforming mypassword	5	High	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 1		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	4	High	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 1		USN-3	As a user ,I can register for application throughphone number.	4	Low	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 1		USN-4	As a user, I can register for the application throughGmail	3	Medium	Hemanathan D Hemnath S Gurram kusuma Hemamalini D

Sprint 2	Dashboard	USN-6	Once I have loggedin , I can see my dashboard.	6	Medium	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 2	Web access	USN-7	As a customer I canaccess the website to predict the weather conditions.	7	High	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 2	Prediction	USN-8	As a customer whenI enter the weather details the website should predict the approximate weather conditions.	7	High	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 3	Analysis	USN -9	As a customer, I wish to store my prediction and makeanalysis.	10	Medium	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 3	Security	USN-10	As a customer I expect my data tobe secured.	10	Medium	Hemanathan D Hemnath S Gurram kusuma Hemamalini D
Sprint 4	Databas eAccess	USN- 11	An administrator I should maintain the website. And updatethe website regularly.	20	Low	Hemanathan D Hemnath S Gurram kusuma Hemamalini D

6.2 REPORT FROM JIRA

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	20	6 days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint 2	20	6 days	31 Oct 2022	5 Nov 2022	20	5 Nov 2022
Sprint 3	20	6 days	7 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint 4	20	6 days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING

7.1 FEATURE 1

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "9a67084c",
   "metadata": {},
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "from flask import Flask, request, jsonify, render_template\n",
    "import joblib\n",
    "import requests"
```

```
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 2,
  "id": "92928833",
  "metadata": {},
  "outputs": [],
  "source": [
   "app = Flask(__name__)\n",
   "model = joblib.load(\"power_prediction.sav\")"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 3,
  "id": "6ee28a40",
  "metadata": {},
  "outputs": [],
  "source": [
   "@app.route('/')\n",
   "def home():\n",
   "    return render_template(\"intro.html\")\n",
   "\n",
   "@app.route('/predict')\n",
   "def predict():\n",
   "    return render_template(\"predict.html\")\n",
```

```
    "\n",
    "@app.route('/windapi', methods=['POST'])\n",
    "def windapi():\n",
    "    city = request.form.get('city')\n",
    "    apikey = \"d8484354b9e388875c48dae8d0d09cd1\"\n",
    "    url = \"http://api.openweathermap.org/data/2.5/weather?q=\" + city + \"&appid=\" +
apikey\n",
    "    resp = requests.get(url)\n",
    "    resp = resp.json()\n",
    "    temp = str(resp[\"main\"][\"temp\"]) +\" °C\"\n",
    "    humid = str(resp[\"main\"][\"humidity\"]) + \" %\"\n",
    "    pressure = str(resp[\"main\"][\"pressure\"]) + \" mmHG\"\n",
    "    speed = str(resp[\"wind\"][\"speed\"]) + \" m/s\"\n",
    "    return render_template('predict.html', temp=temp, humid = humid, pressure=pressure, speed
= speed)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "bab06967",
   "metadata": {},
   "outputs": [],
   "source": [
    "@app.route('/y_predict',methods = ['POST'])\n",
    "def y_predict():\n",
    "    x_test = [[float(x) for x in request.form.values()]]\n",
    "    \n",
```

"   prediction = model.predict(x_test)\n",
"   print(prediction)\n",
"   output = prediction[0]\n",
"   return render_template('predict.html',prediction_text = 'The energy predicted is {:.2f} KWh'.format(output))"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "1768a9b9",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
" * Serving Flask app \"__main__\" (lazy loading)\n",
" * Environment: production\n",
"\u001b[31m   WARNING: This is a development server. Do not use it in a production deployment.\u001b[0m\n",
"\u001b[2m   Use a production WSGI server instead.\u001b[0m\n",
" * Debug mode: off\n"
]
},
{
"name": "stderr",
"output_type": "stream",

"text": [
" * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)\n",
"127.0.0.1 - - [23/Oct/2022 21:55:24] \"GET / HTTP/1.1\" 200 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:24] \"GET /static/images/m123.gif HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:24] \"GET /favicon.ico HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:25] \"GET /predict HTTP/1.1\" 200 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:25] \"GET /static/css/main.css HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:25] \"GET /static/css/media.css HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:25] \"GET /static/css/items_grid.css HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:39] \"POST /windapi HTTP/1.1\" 200 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:39] \"GET /static/css/main.css HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:40] \"GET /static/css/media.css HTTP/1.1\" 404 -\n",
"127.0.0.1 - - [23/Oct/2022 21:55:40] \"GET /static/css/items_grid.css HTTP/1.1\" 404 -\n",
"C:\\Users\\SANJAI\\anaconda3\\lib\\site-packages\\sklearn\\base.py:450: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature names\n",
" warnings.warn(\n",
"127.0.0.1 - - [23/Oct/2022 21:56:19] \"POST /y_predict HTTP/1.1\" 200 -\n",
"127.0.0.1 - - [23/Oct/2022 21:56:19] \""
]
},
{
"name": "stdout",
"output_type": "stream",
"text": [
"[191.65743714]\n"
]
},

```
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "GET /static/css/media.css HTTP/1.1\" 404 -\n",
      "127.0.0.1 - - [23/Oct/2022 21:56:19] \"GET /static/css/main.css HTTP/1.1\" 404 -\n",
      "127.0.0.1 - - [23/Oct/2022 21:56:19] \"GET /static/css/items_grid.css HTTP/1.1\" 404 -\n"
     ]
    }
   ],
   "source": [
    "if __name__ == \"__main__\":\n",
    "    app.run(debug=False)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "5c1b2e47",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
```

"language": "python",
"name": "python3"
},
"language_info": {
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"file_extension": ".py",
"mimetype": "text/x-python",
"name": "python",
"nbconvert_exporter": "python",
"pygments_lexer": "ipython3",
"version": "3.9.12"
}
},
"nbformat": 4,
"nbformat_minor": 5
}

7.2 FEATURE 2

{
"cells": [
{
"cell_type": "code",
"execution_count": 1,
"id": "9a67084c",
"metadata": {},

```
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "from flask import Flask, request, jsonify, render_template\n",
    "import joblib\n",
    "import requests"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "92928833",
   "metadata": {},
   "outputs": [],
   "source": [
    "app = Flask(__name__)\n",
    "model = joblib.load(\"power_prediction.sav\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "6ee28a40",
   "metadata": {},
   "outputs": [],
   "source": [
    "@app.route('/')\n",
```

"def home():\n",
"   return render_template(\"intro.html\")\n",
"\n",
"@app.route('/predict')\n",
"def predict():\n",
"   return render_template(\"predict.html\")\n",
"\n",
"@app.route('/windapi', methods=['POST'])\n",
"def windapi():\n",
"   city = request.form.get('city')\n",
"   apikey = \"d8484354b9e388875c48dae8d0d09cd1\"\n",
"   url = \"http://api.openweathermap.org/data/2.5/weather?q=\" + city + \"&appid=\" + apikey\n",
"   resp = requests.get(url)\n",
"   resp = resp.json()\n",
"   temp = str(resp[\"main\"][\"temp\"]) +\" °C\"\n",
"   humid = str(resp[\"main\"][\"humidity\"]) + \" %\"\n",
"   pressure = str(resp[\"main\"][\"pressure\"]) + \" mmHG\"\n",
"   speed = str(resp[\"wind\"][\"speed\"]) + \" m/s\"\n",
"   return render_template('predict.html', temp=temp, humid = humid, pressure=pressure, speed = speed)"
]
},
{
"cell_type": "code",
"execution_count": 4,
"id": "bab06967",
"metadata": {},

"outputs": [],
"source": [
"@app.route('/y_predict',methods = ['POST'])\n",
"def y_predict():\n",
"   x_test = [[float(x) for x in request.form.values()]]\n",
"   \n",
"   prediction = model.predict(x_test)\n",
"   print(prediction)\n",
"   output = prediction[0]\n",
"   return render_template('predict.html',prediction_text = 'The energy predicted is {:.2f} KWh'.format(output))"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "1768a9b9",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
" * Serving Flask app \"__main__\" (lazy loading)\n",
" * Environment: production\n",
"\u001b[31m   WARNING: This is a development server. Do not use it in a production deployment.\u001b[0m\n",
"\u001b[2m   Use a production WSGI server instead.\u001b[0m\n",

```
     " * Debug mode: off\n"
    ]
   },
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     " * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)\n"
    ]
   }
  ],
  "source": [
   "if __name__ == \"__main__\":\n",
   "    app.run(debug=False)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "5c1b2e47",
  "metadata": {},
  "outputs": [],
  "source": []
 }
],
"metadata": {
 "kernelspec": {
```

```
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.12"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

8. TESTING

Test Cases

Wind Speed (m/s)	Wind Direction (°)	Predicted Power Output (KW)
10.5	100.9	2695.02
6.6	290	751.88
30.7	220	3303.57
25.5	45	3595.69
19.1	0	1135.50
14.8	295	3758.29
8.3	180	1524.59
0.5	88	6.82
3.7	325	34.03
35.2	355	3819.80

8.2 USER ACCEPTANCE TESTING

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Predicting the Energy output of wind turbine using weather condition at the time of the releaseto User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	6	4	2	7	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	20	10	5	26	61
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	29	18	12	35	94

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	2	5
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	6	0	0	6
Version Control	2	0	0	2

9. RESULTS

Performance Metrics

The XGB Regressor ML model that we have used here has better performance in speed and accuracy compared to other models. We have compared the performance metrics of 5 models and selected this as the best for the application. The model performed well for all the test cases. The API developed also performed good with no glitches or lag found during the testing phase.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

• Weather Underground Services provide very accurate Historical Weather Data which increased the accuracy of model.
• Website is more convenient to use due to zero storage.
• With Choosing city, Website can accurately predict power output using weather condition.

DISADVANTAGES

• Weather API is paid and the free version provide limited API requests per day.
• Android Website can be deployed on IBM Cloud.
• No free server available on IBM Cloud for deploying Backend.

11.CONCLUSION

In this project, we used Weather Underground services (subsidiary of IBM) to get accurate historical weather data. For merging this data with Windmill data we learned some Data Analysis concepts. We analyzed several ML models, and chose Random Tree Regressor to develop this model. This project gave us deep insight about the Flutter framework. We integrated the app with model and Weather API using REST API and Flask Back-end. The accuracy of Random Tree Regressor model for this project is 85%.

12. FUTURE SCOPE

The further works that can be done in this project is to include more features in model training to study the effect on the output. A long history of data (dataset of more than 3 years) can be used for training for increased accuracy. The application can be upgraded such that the input values from the sensors are directly fed to the model without the user entering it manually. More web pages can be designed so that the user can control more Wind Mill in the same API. Navigation tabs to move across various Wind mills. The dashboard can be made for User Interactive by making it to show real time graph of the prediction and actual power. Diagnosis of wind mill which perform the least can be done remotely.

13. APPANDIX

SOURCE CODE

HTML CODE FOR INTRO

```
<html>
	<head>
	<title>Wind Energy Prediction</title>
	<style>

		.header {
			top:0px;
			margin:0px;
			left: 0px;
			right: 0px;
			position: fixed;
			background: #6c493a;
			color: white;
			overflow: hidden;
			padding-bottom: 30px;
			font-size: 2.25vw;
			width: 100%;
			padding-left:0px;
			text-align: center;
			padding-top:20px;
		}
		.second{
			top:80px;
			bottom:0px;
			margin:0px;
			left: 0px;
			right: 0px;
			position: fixed;
			padding: 0px;
			width: 100%;
			background-image:url(https://c1.wallpaperflare.com/preview/623/531/630/596ca965a2e3f.jpg);
			background-repeat:no-repeat;
			background-size: contain;
		}
		.inside{
			top:80px;
			bottom:0px;
			margin:0px;
			left: 45%;
			right: 0%;
			position: fixed;
			padding-left: 40px;
```

```
                position: fixed;
                padding-left: 40px;
                padding-top:8%;
                padding-right:40px;
                background-color:#F2D19A;
                font-family:Georgia, serif;
                color:black;
                font-size:20px;
                text-align:justify;

            }
            .myButton{
                 border: none;
                 text-align: center;
                 cursor: pointer;
                 text-transform: uppercase;
                 outline: none;
                 overflow: hidden;
                 color: #fff;
                 font-weight: 700;
                 font-size: 12px;
                 background-color: #6c493a;
                 padding: 10px 15px;
                 margin: 0 auto;
                 box-shadow: 0 5px 15px rgba(0,0,0,0.20);
            }
        </style>
        </head>
        <body>

            <div class="header">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
            <div class="second">
                <div class="inside">Renewable energy, such as wind and solar energy, plays an increasing role in the supply of energy worldwide. This trend will continue because glo
However, levels of production of wind energy are hard to predict as they rely on potentially unstable weather conditions present at the wind farm. In particular, wind speed is crucial for e
            <br><br><br>
            <a href="{{url_for('predict')}}"><button type="button" class="myButton" >Want to predict the energy??</button></a>
            </div>



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

FLASK DISPLAYING HIERARCHY

```
In [1]: import numpy as np
        from flask import Flask, request, jsonify, render_template
        import joblib
        import requests

In [2]: app = Flask(__name__)
        model = joblib.load("power_prediction.sav")

In [3]: @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 = "d8484354b9e388875c48dae8d0d09cd1"
            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)

In [4]: @app.route('/y_predict',methods = ['POST'])
        def y_predict():
            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))

In [ ]: if __name__ == "__main__":
            app.run(debug=False)

         * Serving Flask app "__main__" (lazy loading)
         * Environment: production
           WARNING: This is a development server. Do not use it in a production deployment.
           Use a production WSGI server instead.
         * Debug mode: off
         * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

GITHUB & PROJECT DEMO LINK

GIT REPO: https://github.com/IBM-EPBL/IBM-Project-26728-1660034992

DEMO LINK: https://drive.google.com/file/d/1ePVIyD7O13Z_leQS6-Nioe6ZSy0NpD6T/view?usp=share_link