



நான்  
முதுஸ்வன்

உலகை வெல்லும் இளைய தமிழகம்



# PROJECT REPORT



CHENNAI  
INSTITUTE OF TECHNOLOGY

# **PROJECT TOPIC**

## **Real Time Air Quality Monitoring & Weather Forecasting System**

### **Abstract:**

The Real-time Air Quality Monitoring and Weather Forecasting System is a novel project that aims to develop a system for real-time monitoring and forecasting of air quality and weather conditions. The system uses various sensors to monitor air quality parameters such as particulate matter (PM2.5 and PM10), ozone (O<sub>3</sub>), carbon monoxide (CO), nitrogen dioxide (NO<sub>2</sub>), and sulfur dioxide (SO<sub>2</sub>), as well as weather parameters such as temperature, humidity, pressure and wind speed.

The project includes the development of a web-based platform that displays real-time air quality and weather data from various monitoring stations. The platform will also provide users with historical data, air quality indices, and weather forecasts. The system will also have an alert system that notifies users of critical air quality conditions, such as high pollution levels and adverse weather conditions.

The proposed system will be useful for various stakeholders, including government agencies, environmental organizations, and the general public. The system will provide valuable insights into air quality and weather conditions in real-time, enabling users to take necessary precautions to protect their health and well-being.

The project will involve the development of hardware and software components, including sensor networks, data acquisition and processing systems, and web-based applications. The project team will also conduct extensive field testing to validate the accuracy and reliability of the system.

In conclusion, the Real-time Air Quality Monitoring and Weather Forecasting System is an essential project that aims to develop a system for real-time monitoring and forecasting of air quality and weather conditions. The system will provide valuable insights into air quality and weather conditions, enabling users to take necessary precautions to protect their health and well-being.

**Team Name:** Team Perceptron

**College:** Chennai Institute of Technology, Kundrathur, Chennai

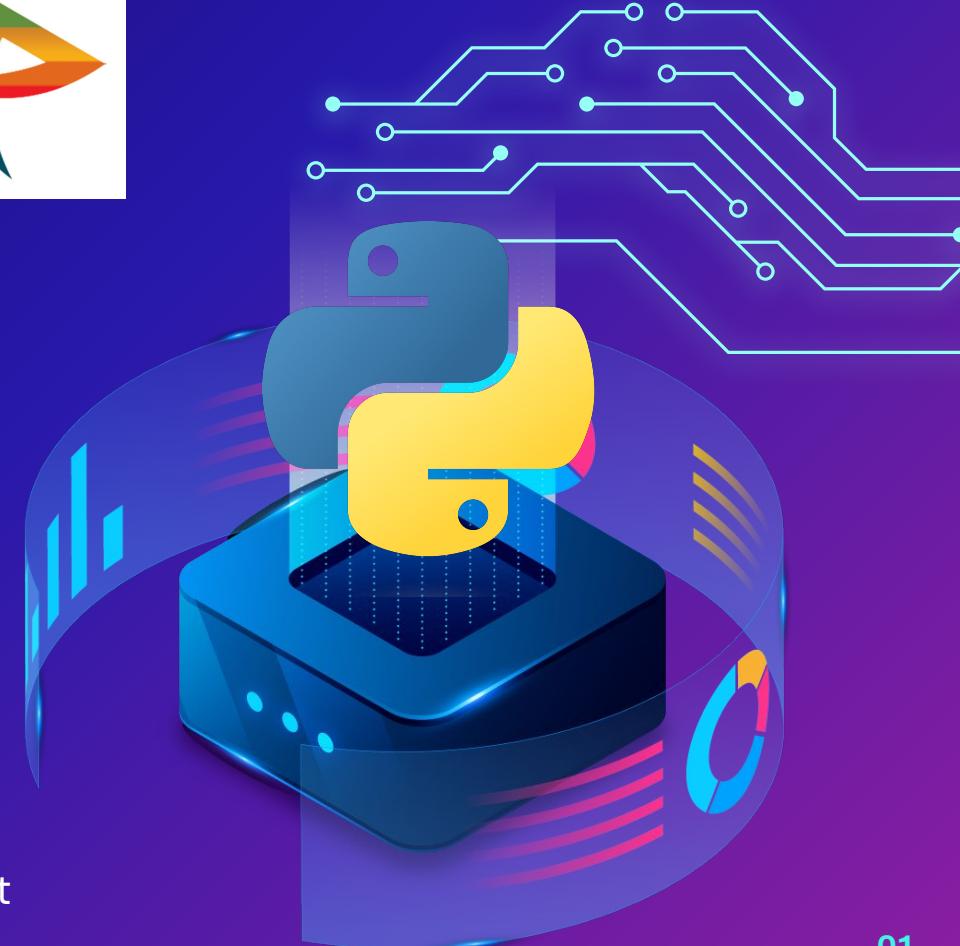
Name	Dept	Email Address	Phone No
Tharun Balaji R	CSE	<a href="mailto:tharunbalajir.cse2021@citchennai.net">tharunbalajir.cse2021@citchennai.net</a>	8248052472
Surya Prakash V	CSE (CyberSec)	<a href="mailto:suryaprakashv.cse2022@citchennai.net">suryaprakashv.cse2022@citchennai.net</a>	9884671340
Nadeem M	CSE (CyberSec)	<a href="mailto:nadeemm.cse2022@citchennai.net">nadeemm.cse2022@citchennai.net</a>	9176766442
Harshithaa RG	IT	<a href="mailto:harshithaarg.it2022@citchennai.net">harshithaarg.it2022@citchennai.net</a>	6379059924

# **PROJECT PRESENTATION**



# HACKATHON PROJECT ABSTRACT

Domain: Machine Learning & Environment



# PROJECT TITLE

Real Time Air Quality Monitoring & Weather Forecasting System

## TEAM PERCEPTRON

- 1) Tharun Balaji R - II yr - CIT Chennai (BE CSE)
- 2) Surya Prakash - I Yr - CIT Chennai (BTech Cybersec)
- 3) Nadeem - I Yr - CIT Chennai (BTech Cybersec)
- 4) Harshitha - I Yr - CIT Chennai (BTech IT)



# TABLE OF CONTENTS

01

## PROBLEM DESCRIPTION

About the problem statement and description

02

## PROJECT BRIEFING

About the project idea and pitching

03

## TECH STACK

Development flow of the projects

04

## PROJECT IMPACT

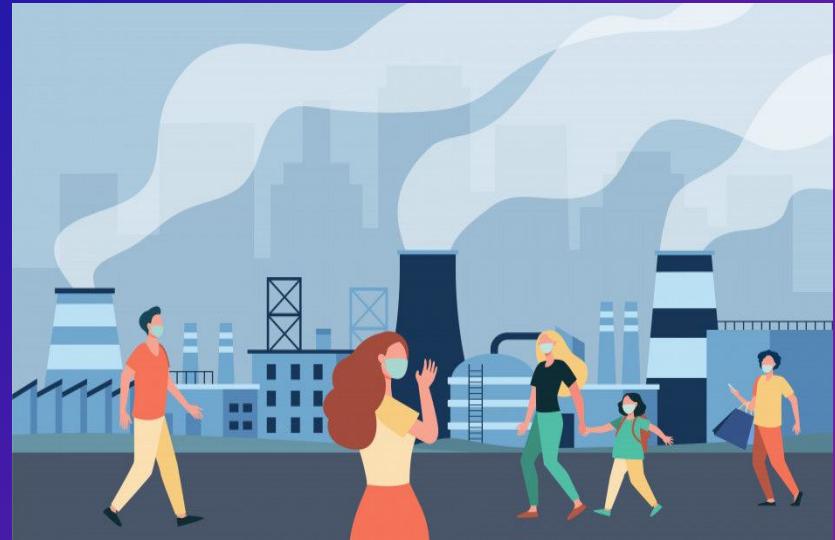
Project applications and use cases



# PROJECT DESCRIPTION

Air pollution represents a serious environmental problem. Each year, millions premature deaths are attributed to air pollution, with huge economic consequences . Furthermore, air pollution is detrimental for ecosystems, damages property, impacts visibility and haze, and threatens food and water security.

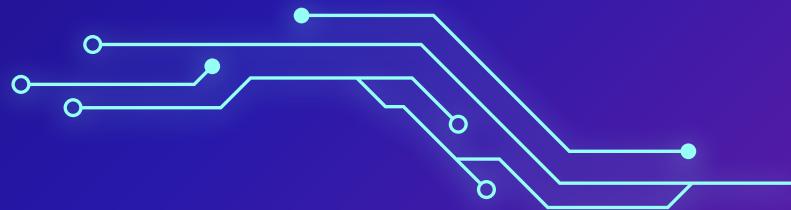
It will provides real-time notifications about prevailing and expected weather conditions which helps governments and local administrations prepare for natural disasters and save lives.



# PROJECT BRIEFING

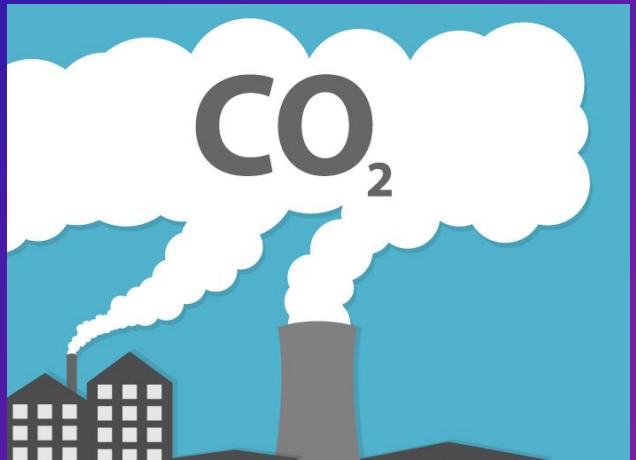


# PROJECT BRIEFING



## PHASE 1: Analysis of Air Quality measurements & Weather forecast data

- ❑ To begin with developing a deployable system, analysis of air quality data such as CO<sub>2</sub> emission index, Particulate matter index, UV radiation, Industrial emissions has to be considered.
- ❑ By diving into these datasets, we have found that these factors play an important role in contaminated air and polluted environment.
- ❑ We analyze the details of these factors for each cities such as Chennai, Bangalore, Hyderabad, Mumbai, etc and designed an analysis chart based on the data to explore more about Air Quality measurement and the causes of environmental issues,



# PROJECT BRIEFING



## PHASE 2: Developing Machine Learning Model

- ❑ Prediction of the air quality is of at most importance. So, building a Machine Learning model that can predict **Real Time** Air Quality & that are reliable and sufficiently accurate, can play an important role as part of an air quality management system.
- ❑ Using Support Vector Machine (**SVM**) or Random Forest Classifier, the contaminated gases from the dataset are segregated and analyzed.
- ❑ Using Data Visualization and Analyzing, the contaminated gases from the dataset are segregated and analyzed.

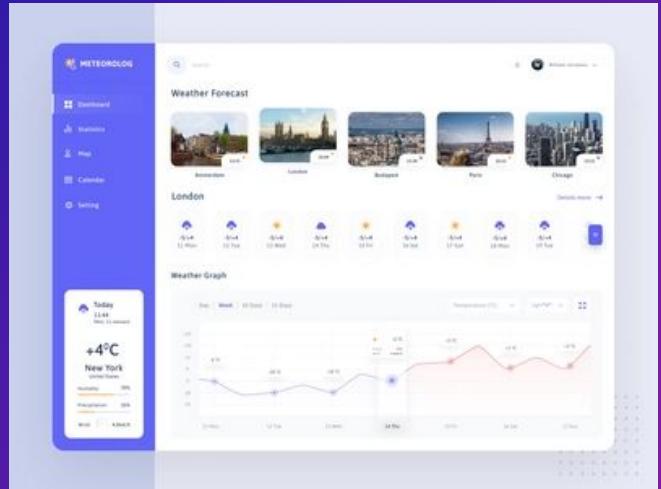


# PROJECT BRIEFING

## PHASE 3: Model Integration with UI

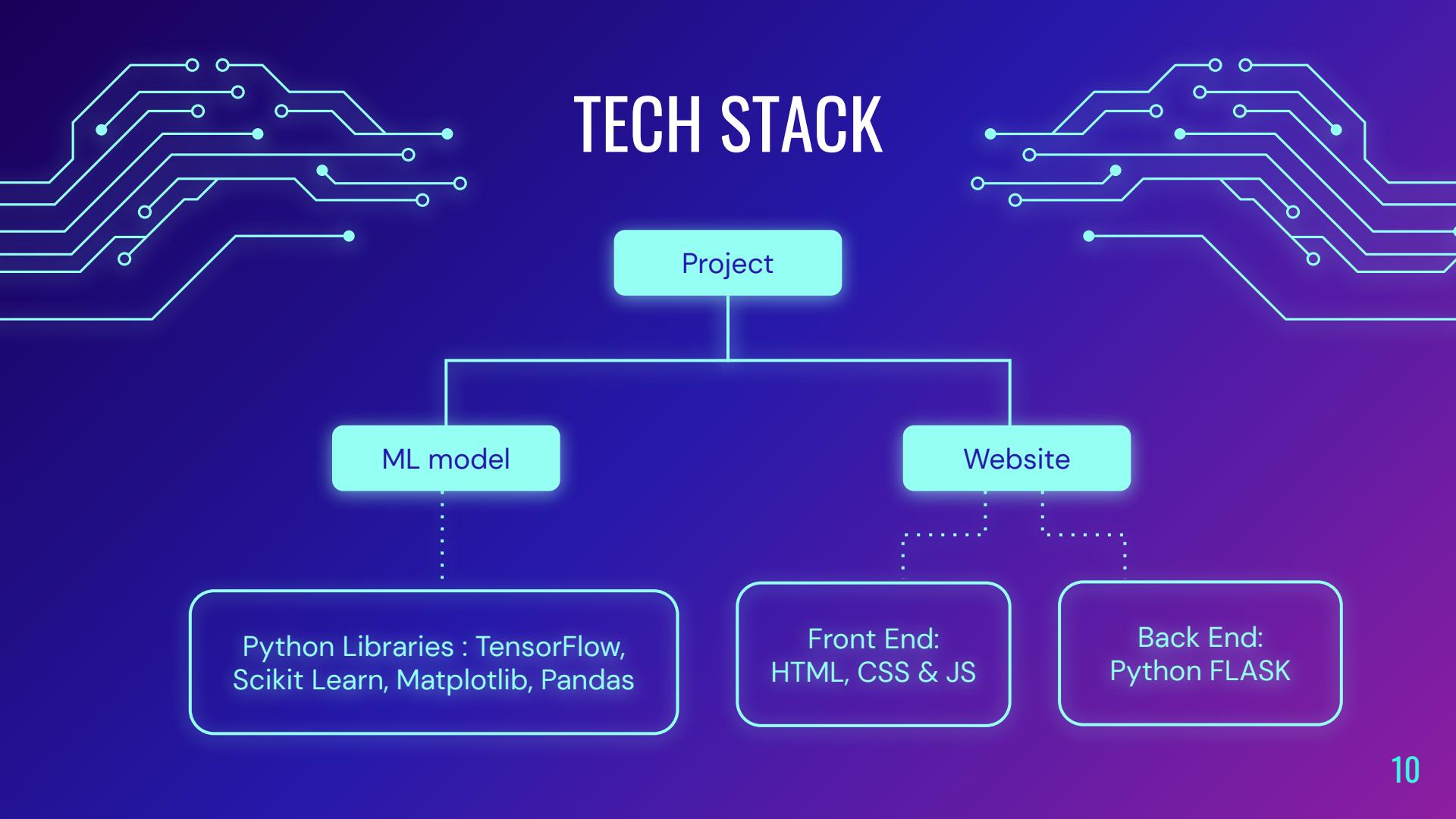


- With the help of WeatherAPI we can get immediate access to local weather conditions and upcoming forecast. It will Providing real-time notifications about prevailing and expected weather conditions which helps governments and local administrations prepare for natural disasters and save lives.
- After completing the model, a sophisticated UI model ([Dynamic website](#)) is used to integrate the ML model and Weather API to display the prediction for the user.

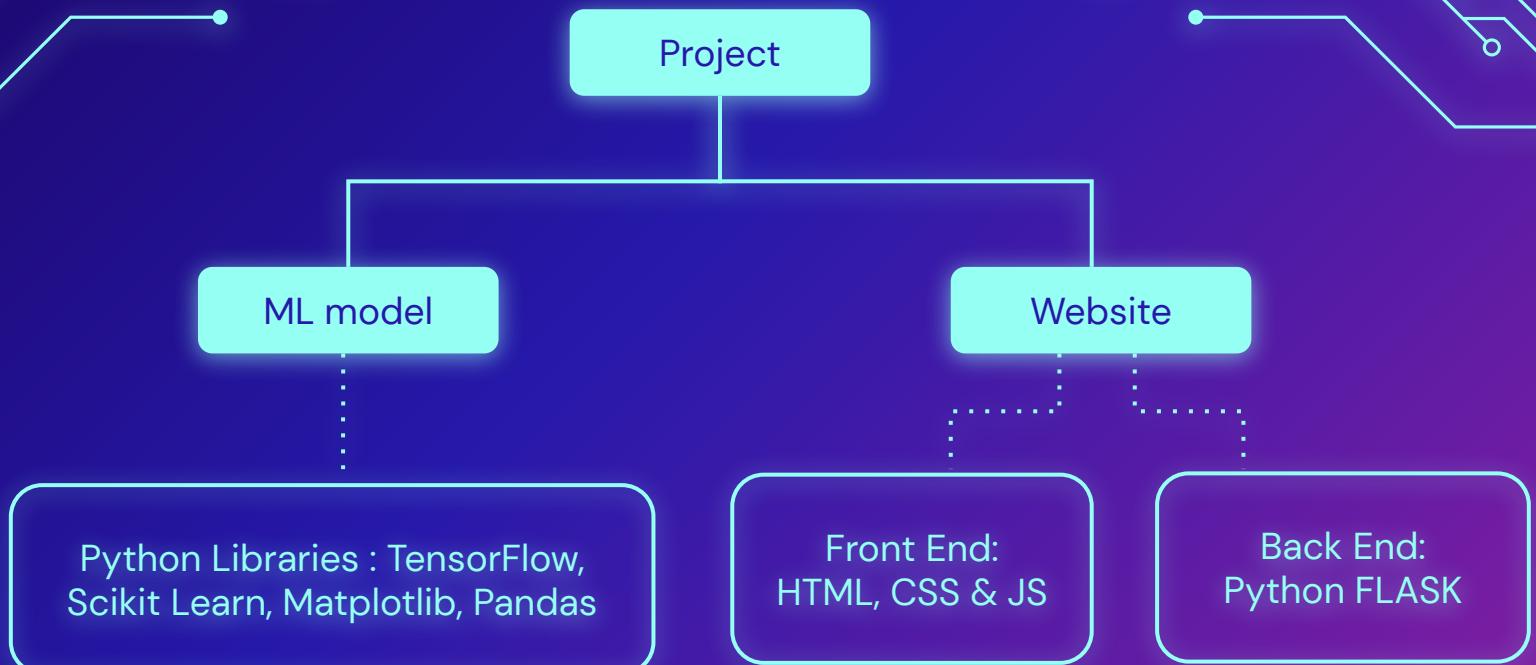


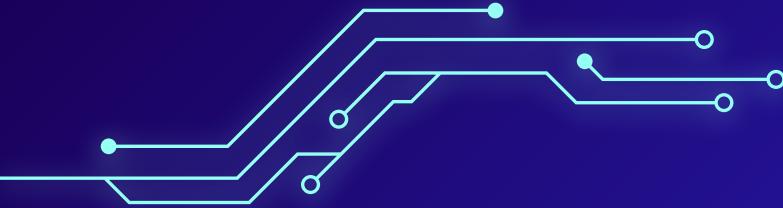


# PROJECT TECH STACK



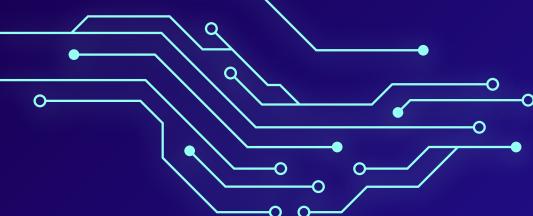
# TECH STACK





# PROJECT IMPACT



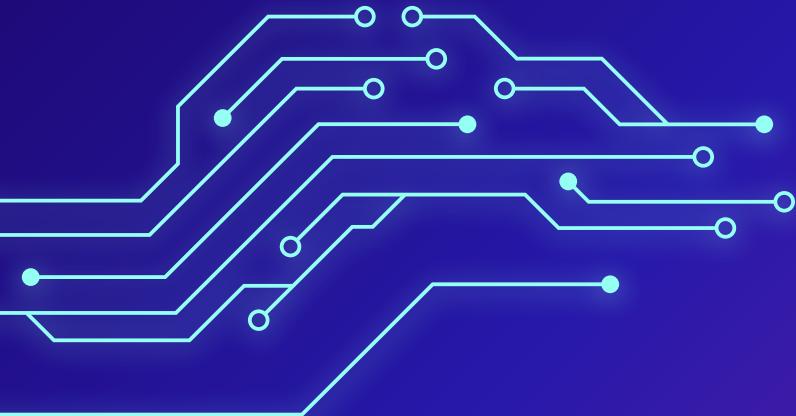


# PROJECTS IMPACT



On a larger extent, The **air quality (AQ)** Forecast lets the public know expected air quality conditions for **next 72 hours** so that Government authorities can take action to manage the air quality and issue health advisories. With the help of WeatherAPI we can get immediate access to local weather conditions and upcoming forecast, It will Providing real-time notifications about prevailing and expected weather conditions, which **helps governments** and local administrations prepare for natural disasters and save lives.

# THANK YOU



# PROJECT INSIGHTS

## Project Description:

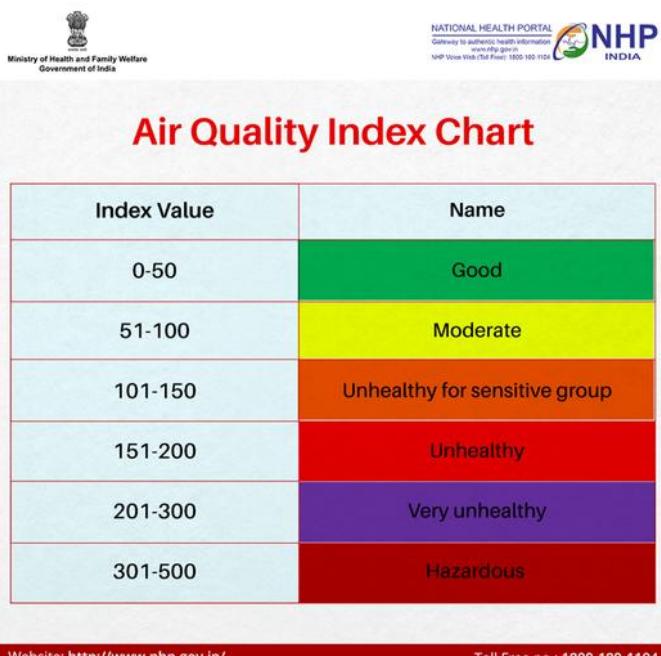
The Real-Time Air Quality Monitoring & Weather Forecasting System is a project that employs HTML, CSS for front-end, Python **Flask** for back-end, and **OpenWeather** API for fetching live weather forecasting data and air quality values for any city in Tamil Nadu. The project utilizes a machine learning model based on **Random Forest** Classifier, which has achieved an accuracy rate of **90%** and minimal error percentage.

The project directly forecasts weather data fetched from the API and provides daily, hourly, and monthly order weather forecasts. For air quality, the project fetches data about air pollutant values such as **CO, NO, NO<sub>2</sub>, O<sub>3</sub>, PM2.5, PM10**, etc. The project predicts the Air Quality Index (**AQI**) value and provides recommendations based on the prediction. The system is designed to provide valuable insights into air quality and weather conditions in real-time, enabling users to take necessary precautions to protect their health and well-being.

## Project Tech Stack used:

Front End	Back End	Machine Learning
HTML5, CSS  	Python Flask  	Numpy, Pandas, Random Forest Classifier   

## AQI Index Chart:



## Project Code:

GITHUB Repository Link - [Click Here](#)

## Project Deploy:

WEBSITE Name - AirCast  
Link - <https://aircast.onrender.com>

## API Links:

- 1) [OpenWeather Weather Forecast API](#)
- 2) [OpenWeather Air Quality Data API](#)
- 3) [OpenWeather Geo Coding API](#)

# **PROJECT SCREENSHOTS**

## 1) Project NoteBook Hierarchy Structure

The screenshot shows a Jupyter Notebook titled "tn\_airquality\_analysis.ipynb" in Visual Studio Code. The left sidebar displays the file structure of the project, including "dataset", "flask-backend", "frontend", "tobadded", "air\_quality\_analysis.ipynb", "air-quality-analysis-edu-and...", "airquality\_predictor.py", "airquality\_script.py", "main.ipynb", "ml\_model\_creation.ipynb", "README.md", and "tn\_airquality\_analysis.ipynb". The main area of the notebook contains the following sections:

- Importing Libraries**  
1 cell hidden ...
- Feeding input dataset**  
2 cells hidden ...
- Exploratory Data Analysis**  
2 cells hidden ...
- Data Visualization**  
4 cells hidden ...
- Preprocessing**  
6 cells hidden ...
- Classification**

At the bottom of the notebook, there are buttons for "Run Testcases", "Connect", and "Git Graph". The status bar at the bottom right indicates "Cell 21 of 28", "Go Live", and other connectivity icons.

## 2) Project Data Visualization

The screenshot shows a Jupyter Notebook titled "tn\_airquality\_analysis.ipynb" in Visual Studio Code. The left sidebar displays the same project structure as the previous screenshot. In the main area, a code cell contains the following Python code:

```
plt.show()
plotting('Overall Pollution Content')
```

This code generates a line plot titled "Overall Pollution Content" showing pollution levels over time. The x-axis is labeled "Chennai City" and ranges from 0 to 100k. The y-axis is labeled "Overall Pollution Content" and ranges from 0 to 12k. The plot shows a highly volatile orange line with several sharp peaks, notably around 10k and 20k on the x-axis.

Below the plot, another code cell shows the first five rows of a DataFrame:

```
df1.head()
```

The resulting table is as follows:

	date	aqi	co	no	no2	o3	so2	pm2_5	pm10	nh3	city
0	1640995200	4	440.60	0.00	9.60	77.25	11.21	44.11	55.34	1.65	Chennai
1	1640998800	4	453.95	0.00	10.20	79.39	11.44	45.22	57.18	1.74	Chennai
2	1641002400	4	560.76	0.02	14.91	77.25	13.23	46.26	59.88	2.60	Chennai
3	1641006000	4	634.19	0.32	19.54	76.53	15.26	47.94	63.51	2.72	Chennai
4	1641009600	5	547.41	0.54	13.71	92.98	18.12	53.48	70.75	0.65	Chennai

### 3) Project Data Preprocessing

The screenshot shows a Visual Studio Code interface with a Python notebook file open. The code cell contains:

```
cor = df1.corr()
cor.style.background_gradient(cmap='coolwarm')
```

A warning message is displayed:

```
C:\Users\tharu\AppData\Local\Temp\ipykernel_16404\3030549476.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or
```

Below the code is a correlation heatmap for the dataset:

	date	aqi	co	no	no2	o3	so2	pm2.5	pm10	nh3	city_encoded
date	1.000000	0.015263	0.080210	0.027644	0.046685	0.031825	0.012919	0.126014	0.123825	0.031753	-0.000000
aqi	0.015263	1.000000	0.477915	0.167133	0.369378	0.325941	0.274881	0.797283	0.801661	0.221678	-0.106643
co	0.080210	0.477915	1.000000	0.748588	0.755250	-0.245072	0.336509	0.683381	0.696849	0.570854	-0.259258
no	0.027644	0.167133	0.748588	1.000000	0.425175	-0.186453	0.359746	0.370584	0.380455	0.397239	-0.164643
no2	0.046685	0.369378	0.755250	0.425175	1.000000	-0.354896	0.588148	0.494675	0.513397	0.520171	-0.339488
o3	0.031825	0.325941	-0.245072	-0.186453	-0.354896	1.000000	0.065809	0.190579	0.161482	-0.304737	0.075261
so2	0.012919	0.274881	0.336509	0.359746	0.588148	0.065809	1.000000	0.325449	0.314127	0.099255	-0.247682
pm2.5	0.126014	0.797283	0.683381	0.370584	0.494675	0.190579	0.325449	1.000000	0.993912	0.304405	-0.124665
pm10	0.123825	0.801661	0.696849	0.380455	0.513397	0.161482	0.314127	0.993912	1.000000	0.350644	-0.135343
nh3	0.031753	0.221678	0.570854	0.397239	0.520171	-0.304737	0.099255	0.304405	0.350644	1.000000	-0.223639
city_encoded	-0.000000	-0.106643	-0.259258	-0.164643	-0.339488	0.075261	-0.247682	-0.124665	-0.135343	-0.223639	1.000000

Below the heatmap is more code:

```
y = df1["aqi"]
x = df1[['co','no','no2','o3','so2','pm2.5','pm10','nh3','city_encoded']]
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)

print('Classes and number of values in trainset',Counter(y_train))
```

Output of the print statement:

```
... Classes and number of values in trainset Counter({0: 18867, 3: 17807, 1: 15762, 4: 15105, 2: 12098})
```

### 4) Project Model Training using SVM, RFC and XGBC Classifier (RFC chosen)

The screenshot shows a Visual Studio Code interface with a Python notebook file open. The code cell contains:

```
cls=SVC()
cls.fit(X_train,y_train)
svmpred=cls.predict(X_test)

cm=confusion_matrix(y_test,svmpred)
print("confussion matrix")
print(cm)
print("\n")
accuracy=accuracy_score(y_test,svmpred)
print("accuracy",accuracy*100)
```

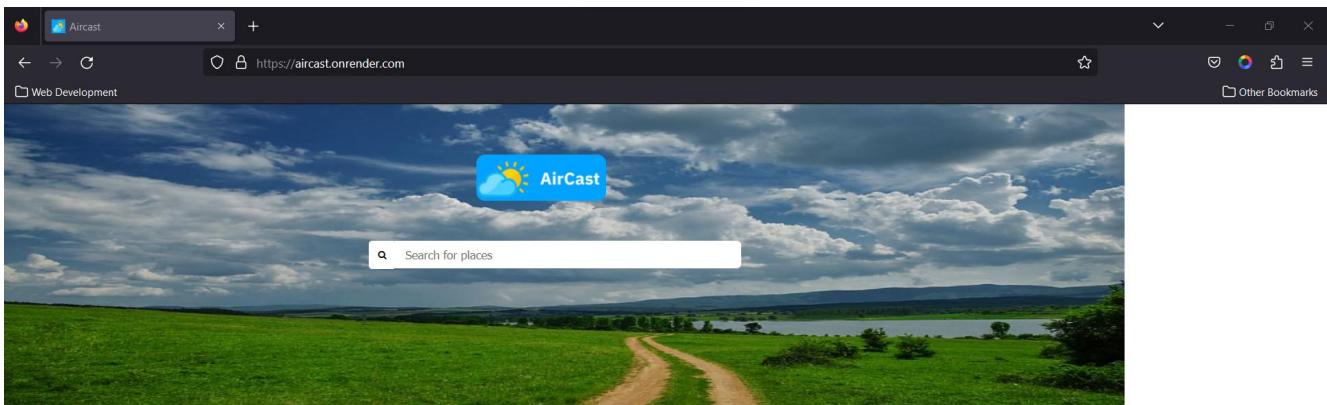
Below the SVM code is a section titled "Random Forest Classifier (RFC)" containing:

```
rf = RandomForestClassifier(n_estimators=20, random_state=25)
rf.fit(X_train, y_train)
rf_predict=rf.predict(X_test)
rf_predict1=rf.predict(X_train)
rf_conf_matrix = confusion_matrix(y_test, rf_predict)
rf_acc_score = accuracy_score(y_test, rf_predict)
print("confussion matrix")
print(rf_conf_matrix)
print("\n")
print("accuracy",rf_acc_score*100)
```

Output of the confusion matrix:

```
... confusion matrix
[[8176 24 0 0 0]
 [ 81 6085 498 0 0]
 [ 0 158 3766 1177 0]
 [ 0 0 1223 5846 689]
 [ 0 0 0 376 6032]]
```

## 5) Developed UI part

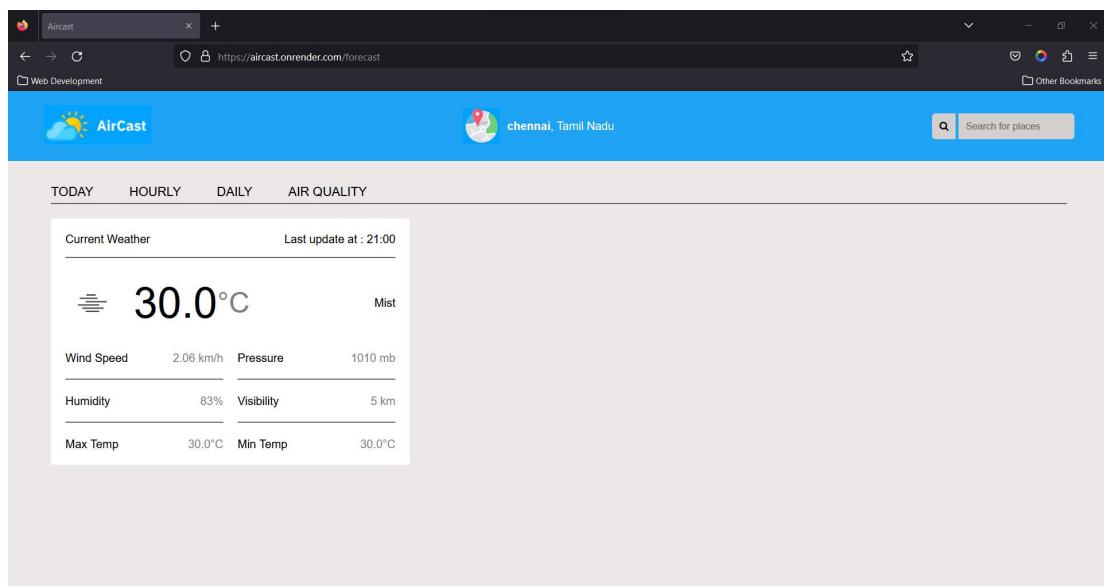


## 6) Integrated Backend with Flask

```
File Edit Selection View Go Run Terminal Help
EXPLORER: ...
main.py
flask-backend > main.py > ...
1 from flask import Flask, render_template, session, redirect, url_for, request
2 import os
3 import requests
4 import requests, json
5 from dotenv import load_dotenv
6 from datetime import datetime
7 from time import strftime, localtime
8 import pickle, numpy as np
9 model = pickle.load(open("airquality_model.pkl","rb"))
10
11 i = 0
12 dt = datetime.now()
13 app = Flask(__name__)
14 load_dotenv()
15 api = os.getenv("api")
16 Flask.secret_key = os.getenv("secret_key")
17
18 def hourlydata():
19     my_var = session.get("my_var", None)
20     if my_var is None:
21         my_var = ["Chennai"]
22     list_of_data = requests.get(
23         f"https://api.openweathermap.org/data/2.5/weather?q={my_var[0]},tn,in&appid={api}&units=metric"
24     ).json()
25     data = requests.get(
26         f"https://pro.openweathermap.org/data/2.5/forecast/hourly?q={my_var[0]}&cnt=10&appid={api}&uni"
27     ).json()
28     data = {
29         "cityname": f"{my_var[0]}",
30         "country_code": str(list_of_data["sys"]["country"])
31     }
```

The screenshot shows the Visual Studio Code interface with the "main.py" file open in the center. The code is written in Python and uses the Flask framework to interact with an OpenWeatherMap API to get weather data for Chennai, India. It also loads a local "airquality\_model.pkl" file. The code includes imports for Flask, requests, and pickle, along with various utility functions and classes.

## 7) Fully functioning Dynamic Website (Weather Forecasting - Today)



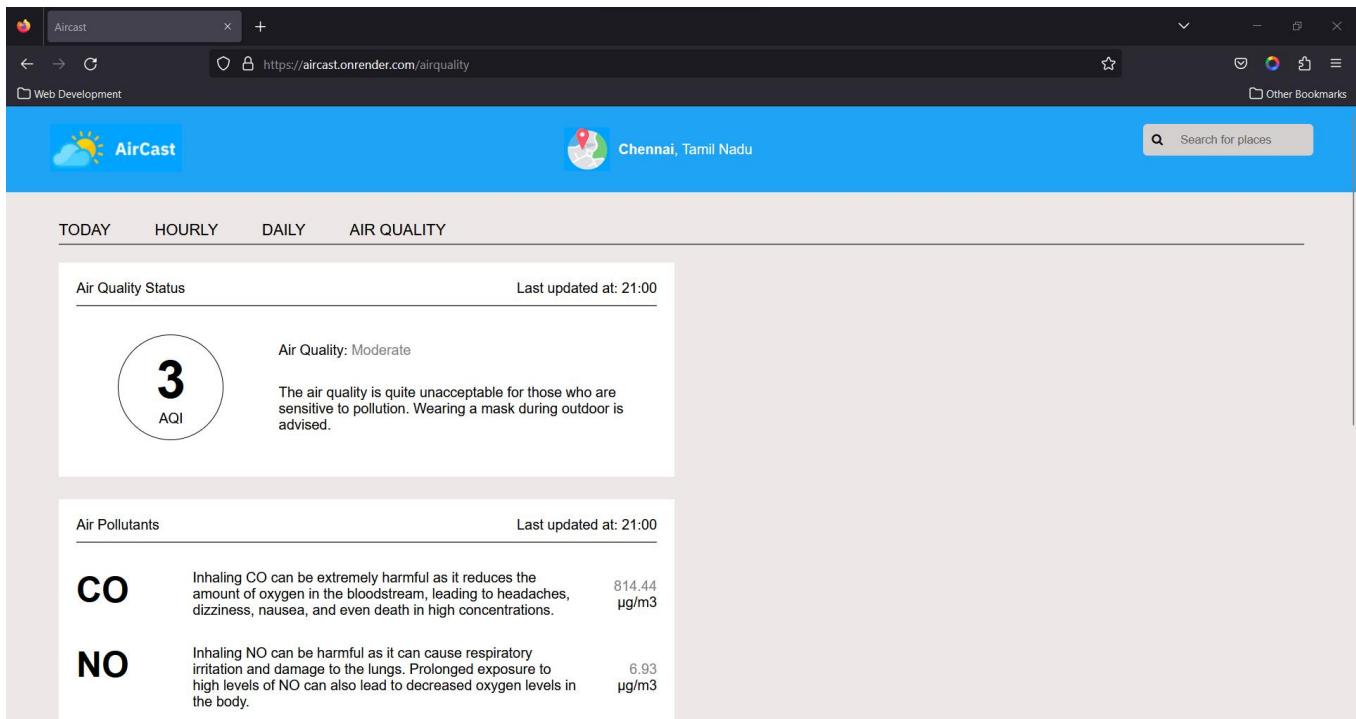
## 8) Fully functioning Dynamic Website (Weather Forecasting - Hourly)

The screenshot shows a Firefox browser window displaying the AirCast website at <https://aircast.onrender.com/hourly>. The page is titled "Aircast" and shows weather data for "chennai, Tamil Nadu". The interface includes tabs for "TODAY", "HOURLY", "DAILY", and "AIR QUALITY". The "HOURLY" tab is selected, showing two data cards for 03:00:00 and 04:00:00. Each card displays the temperature (29.9°C and 30.5°C respectively), weather condition (Light Rain), and other metrics like Wind Speed (3.11 km/h), Pressure (1009 mb), Humidity (67%), and Visibility (10 km). The last update time is listed as 21:00 for both entries.

## 9) Fully functioning Dynamic Website (Weather Forecasting - Daily)

The screenshot shows a Firefox browser window displaying the AirCast website at <https://aircast.onrender.com/daily>. The page is titled "Aircast" and shows weather data for "chennai, Tamil Nadu". The interface includes tabs for "TODAY", "HOURLY", "DAILY", and "AIR QUALITY". The "DAILY" tab is selected, showing four data cards for the dates 02-05-2023, 03-05-2023, 04-05-2023, and 05-05-2023. Each card displays the temperature (27.0°C, 29.9°C, 30.1°C, and 30.1°C respectively), weather condition (Moderate Rain for 02-05-2023 and Light Rain for the others), and the last update time (all at 21:00).

## 10) Fully functioning Dynamic Website (Air Quality Prediction)



### Project Keyvalues:

- The project forecasts live Weather forecasting using OpenWeather API, as mentioned in project Insights, for any location in Tamil Nadu state or any other Indian states. Since the location of the user input city will be converted to corresponding latitude and longitude of that city by Geo Coding OpenWeather API. The results are fetched by the website.
- The project contains a well trained ML model for Air Quality Prediction and currently it has been trained for Tamil Nadu's top 10 cities: Chennai, Coimbatore, Madurai, Tiruchirappalli, Salem, Tirunelveli, Erode, Thoothukudi, Dindigul with an accuracy of 90% and can predict Air Quality Index (AQI) precisely by using the values of air pollutants fetched from API (CO,NO,PM2.5,PM10,etc) as per Indian Government standard.
- The project is equipped with a Recommendation System and Weather related alerts which lets the public know expected air quality conditions for next 72 hours so that Government authorities can take action to manage the air quality and issue health advisories.
- The project can also be integrated with other applications such as Health & Fitness Tracker, Pollution Tracker, etc which enables users to take necessary precautions to protect their health, well-being and reschedule their work.

**THANK  
YOU**