NAME: **BHARATHI M**

COLLEGE CODE: **4101**

DEPT: **CSE- 3 YEAR**

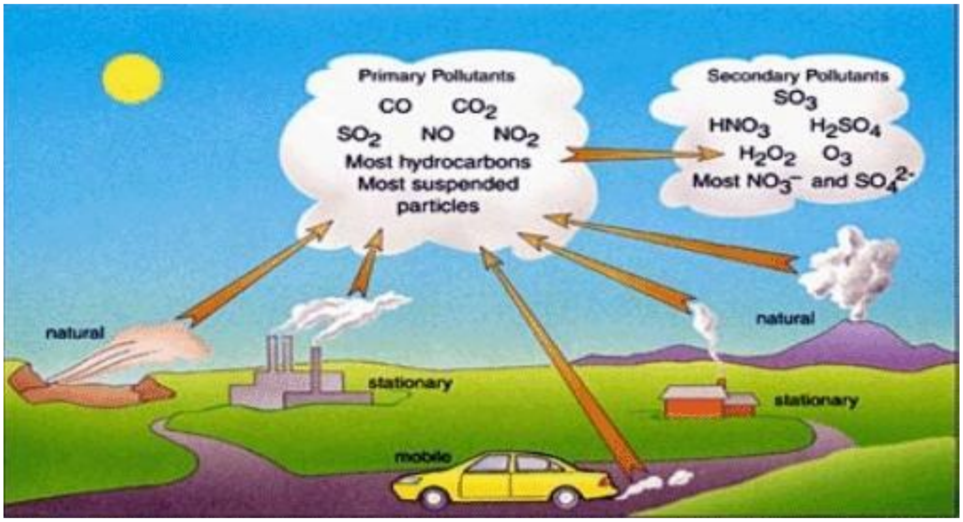
DOMAIN: **DATA ANALYTICS WITH COGNOS**

PROBLEM STATEMENT: **AIR QUALITY ANALYSIS**

PROJECT PHASE 1 :

AIR QUALITY ANALYSIS

Humans can survive for 30 days without eating, 3 days without drinking, yet only 3 minutes without breathing. Of course our need for air is also constant, we rely on it at all times indoors and outdoors although can often be less clean than we would hope.



Abstract

Air quality analysis using Python is a powerful tool for understanding and predicting air quality patterns. Python is a general-purpose programming language that is well-suited for data analysis and visualization. There are a number of Python libraries available for air quality analysis, including:

pandas: A library for data manipulation and analysis.

matplotlib: A library for data visualization.

seaborn: A library for statistical graphics.

scikit-learn: A library for machine learning.

Project Steps

Phase 1: Project Definition and Design Thinking

Project Definition: The project aims to analyze and visualize air quality data from monitoring stations in Tamil Nadu. The objective is to gain insights into air pollution trends, identify areas with high pollution levels, and develop a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels. This project involves defining objectives, designing the analysis approach, selecting visualization techniques, and creating a predictive model using Python and relevant libraries.

Design Thinking:

1. Project Objectives: Define objectives such as analyzing air quality trends, identifying pollution hotspots, and building a predictive model for RSPM/PM10 levels.

☆Our goal is to analyze the quality of the air from the given data set .

● Collecting or downloading the data from the given dataset.

Dataset

Link: https://tn.data.gov.in/resource/location-wise- daily-ambient-air-quality-tamil-nadu-year-2014

● Analyze the data .

● And then from the given data set our aim is to find the most polluted city and to build a predictive model.

```
import pandas as pd

df = pd.read_csv('csv_file_path’)

# Calculate the metric

city = df.groupby('City/Town/Village/Area')[‘SO2’].sum().reset_index()

# Find the most affected city

most_affected_city = city.loc[city[‘SO2'].idxmax()]

print("The most affected city is:", most_affected_city['City/Town/Village/Area'])
```

It results the most polluted city in given dataset only by taking the values of SO2 .

We can also check for other columns also such as NO2 ,RSPM/PM10 ,etc..

Machine Learning :

As we need to make predictions , consider building machine learning models.

Split the data into training and testing sets, train models, and evaluate their performance.

2. Analysis Approach: Plan the steps to load, preprocess, analyze, and visualize the air quality data.

Data Collection:

Downloading the dataset in the CSV format using the above mentioned link.

Data Loading:

Using libraries like Pandas in Python to load the data into a structured format like a DataFrame. And also specifying the path of the csv file.

```
csv_file_path = 'weather.csv'

df = pd.read_csv(csv_file_path)
```

Data Preprocessing:

Handle missing data: Decide on a strategy (e.g., imputation or removal) for dealing with missing values.

Data cleaning: Correct any inconsistencies or errors in the data.

```
# df = df.dropna()

# df = df.drop_duplicates()
```

Data visualization: Create plots and charts (histograms, scatter plots, box plots) to explore data distributions and relationships between variables.

Use libraries like Matplotlib, Seaborn, or Plotly in Python for visualization.

3.Visualization Selection: Determine visualization techniques (e.g.,line charts, heatmaps) to effectively represent air quality trends and pollution levels.

Different visualization methods can highlight various aspects of the data, so it's important to choose the right ones based on your specific goals and the characteristics of your data. Here are some visualization techniques you can consider:

Line Charts: Line charts are excellent for displaying time-series data, such as air quality trends over a specific period.

Bar Charts: Bar charts can be used to compare air quality measurements across different locations or categories. For example, you can create a bar chart to compare pollution levels in different cities or display pollutant concentrations for various pollutants side by side.

Heatmaps: Heatmaps are useful for visualizing spatial data.

Pie Charts: Each slice of the pie can represent a different pollutant, and the size of the slice corresponds to its contribution to the total pollution.

Hence here I prefer using bar charts to visualize the pollution rate in different cities based on RSPM and PM10 levels

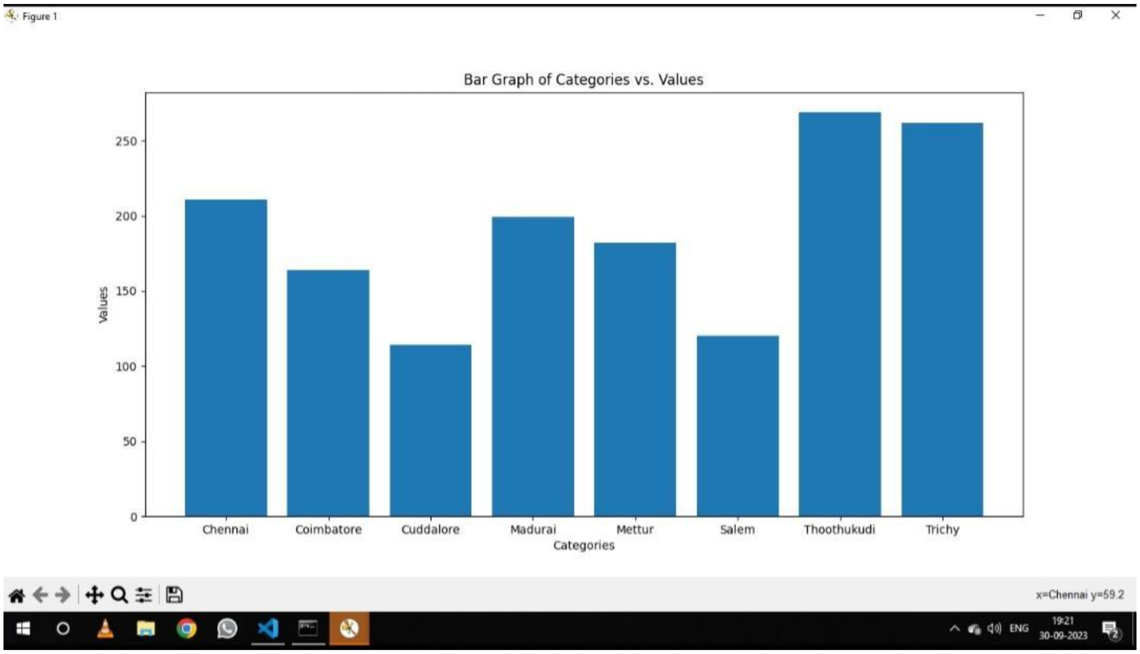
The code for the following Design Thinking:

```
import pandas as pd

import matplotlib.pyplot as plt

# Specify the path to your CSV file

csv_file_path = 'weather.csv'

# Read the CSV file into a DataFrame
```

```
df = pd.read_csv(csv_file_path)

# Extract data for plotting

categories = df['City/Town/Village/Area']

values = df['RSPM/PM10']

# Create a bar graph

plt.bar(categories, values)

# Add labels and a title

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Bar Graph of Categories vs. Values')

# Display the graph

plt.show()
```

OUTPUT:

CONCLUSION:



The air quality analysis module is essential for understanding the state of air pollution, its impact on public health, and the effectiveness of air quality management measures. Advancements in sensor technology, data analytics, and modeling techniques continue to enhance the accuracy and timeliness of air quality assessments, contributing to better-informed decisions for environmental and public health management.