Real-Time Air Quality Monitoring & Weather Forecasting System

MachineLearners

Sharmistha Das(Team Leader)

College: **Narula Institute of Technology**
Stream: **IT**
Year of graduation: **2024**

Agniban Saha

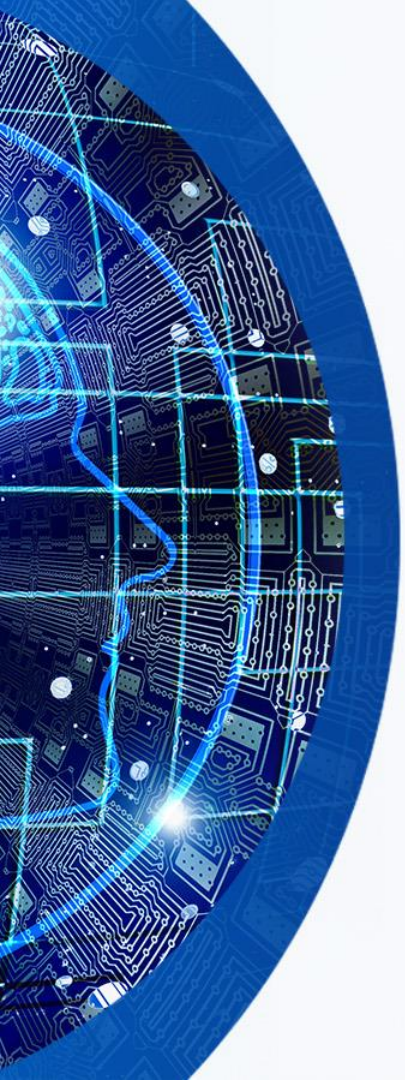
College: **Narula Institute of Technology**
Stream: **ECE**
Year of graduation: **2024**

Deep Sarkar

College: **Narula Institute of Technology**
Stream: **IT**
Year of graduation: **2024**

Abhirup Basak

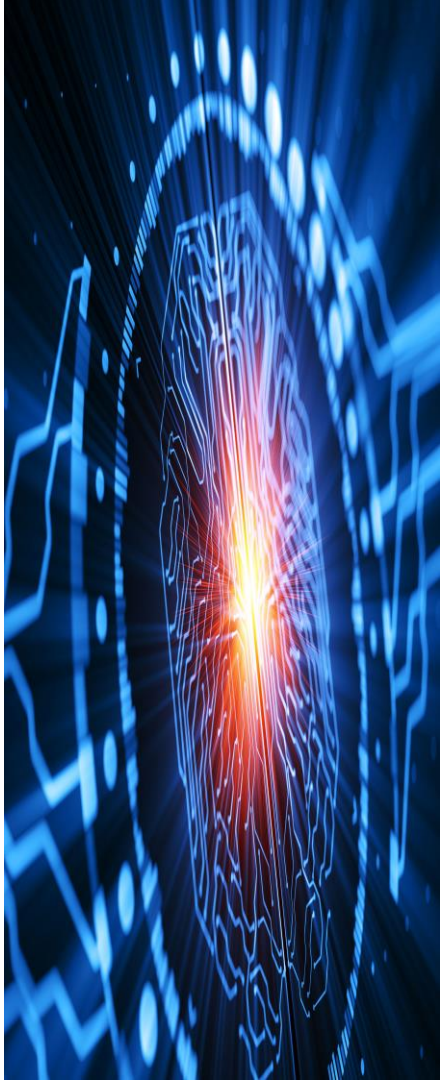
College: **Narula Institute of Technology**
Stream: **IT**
Year of graduation: **2024**



ROLE OF TEAM MEMBERS

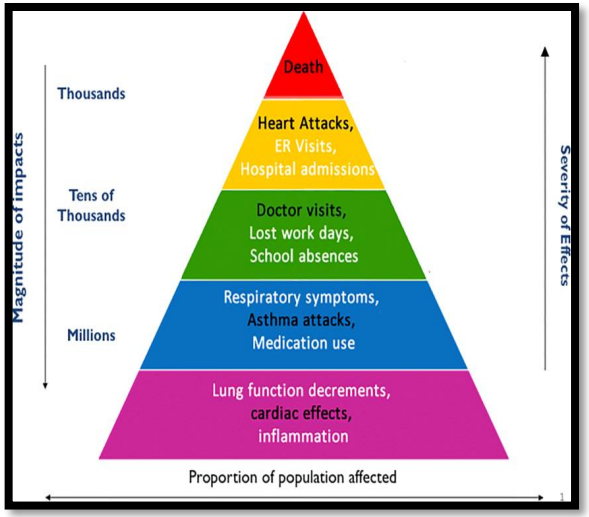
Sharmistha Das Team Leader	Agniban Saha Team Member1	Deep Sarkar Team Member2	Abhirup Basak Team Member3
1. Pre-Requisites	1. Pre-Requisites	1. Pre-Requisites	1. Pre-Requisites
2. Video Presentation	2. Building the Solution	2. Preparing UI/UX Design	2. Preparing UI/UX Design
3. Solution Submission	3. Preparing Documentation	3. Building the Solution	3. Building the Solution

CONTENTS



PROJECT DESCRIPTION

- ✓ Air pollution represents a serious environmental problem. Air quality forecasting is a vital tool for local health and air managers to make informed decisions on mitigation measures to reduce public exposure risk.
- ✓ Air quality forecasts, if they are reliable and sufficiently accurate, can play an important role as part of an air quality management system.
- ✓ The air quality (AQ) forecast lets the public know expected air quality conditions for the next 72 hours so that government authorities can take action to manage the air quality and issue health advisories.
- ✓ With the help of Weather-API, we can get immediate access to local weather conditions and an upcoming forecast. Providing real-time notifications about prevailing and expected weather conditions helps governments and local administrations prepare for natural disasters and save lives.



Proposed Methodology

- The air quality and meteorological datasets have been scrapped from : https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india
- The collected data is then modified, as we have calculated the daily hours of air quality data into daily average data for target value .
- The processed data is then divided into 2 groups: Training set and Testing set, and then fed into the Random Forest Regressor.
- To show the performance of our proposed idea, we trained the air quality data and forecasted it by fitting models with optical hyper-parameters. The forecasted values were then compared with the observed values.
- The performance is calculated in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).
- Then we integrate our developed learnings to a Web-based application so that the user could also be alerted about the air quality index, and suggestions on how to react and improve air quality can be disseminated quickly by a mobile.

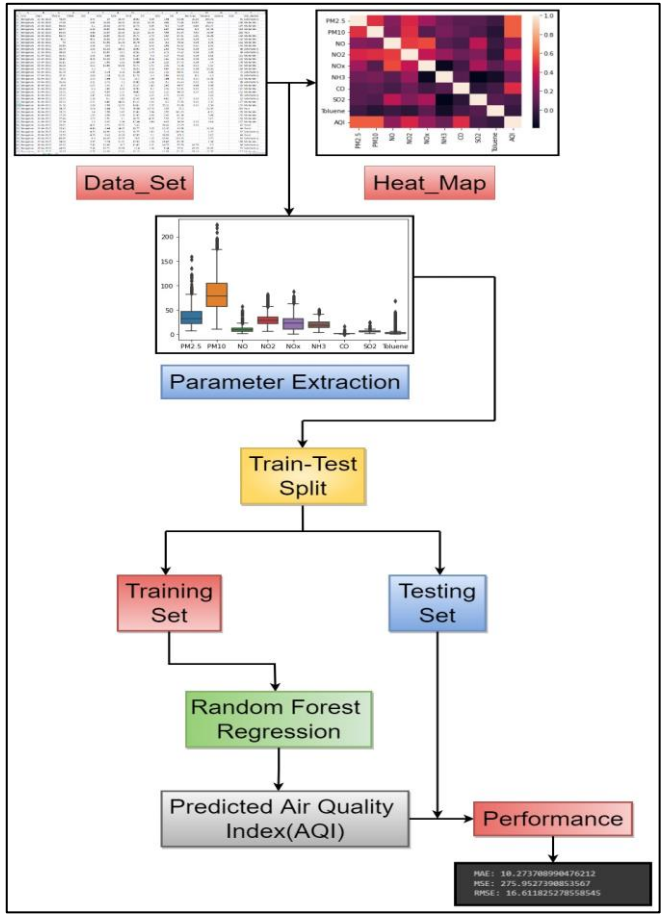
Technology Stack

- Air Quality Dataset with Parameters
- Machine Learning Concepts-
  - ✓ Random Forest Regression
- Graphs-
  - ✓ Pair Plot
  - ✓ Scatter Plot
  - ✓ Box Plot
- Heat Map –
  - ✓ Seaborn Heat Map
- Python Programming Language
- Flutter
- Weather APK

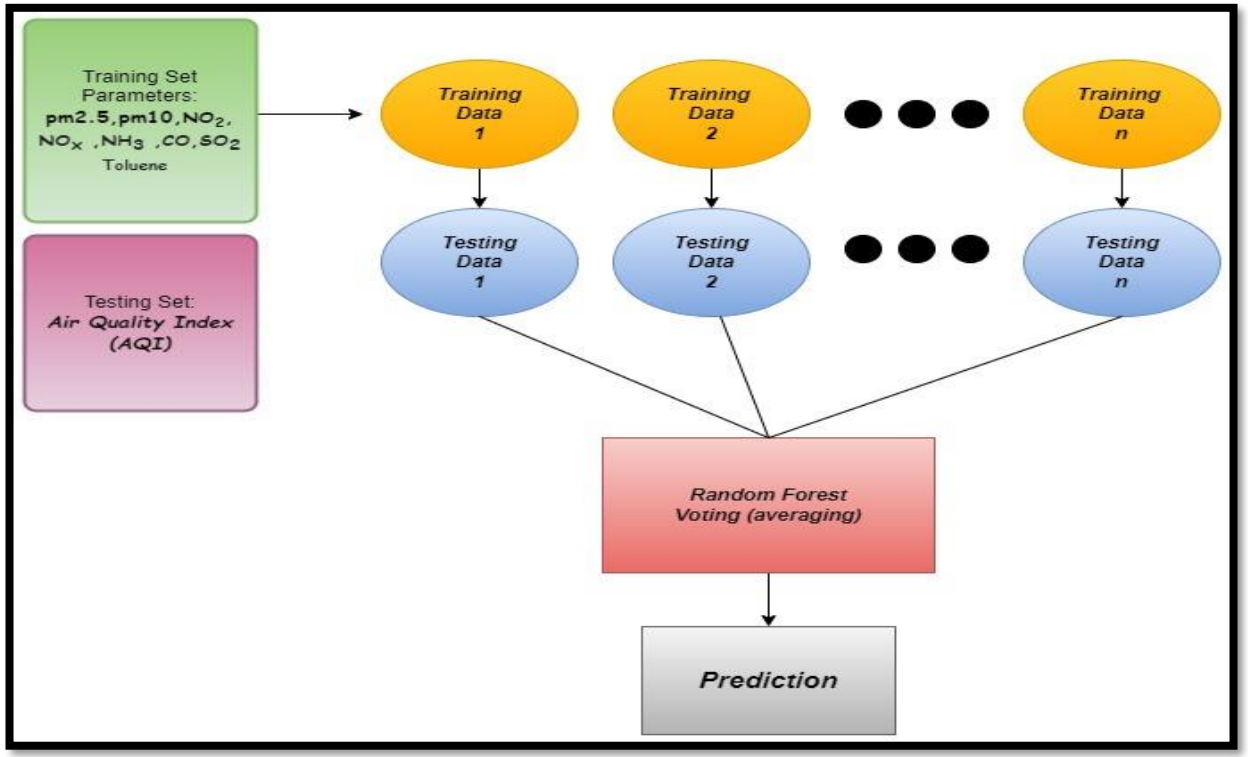
Use Cases

- Tracking air pollution.
- Forecasting dust storms
- Tracking global wildfires
- Airlines
- Research Environment
- Health Sector

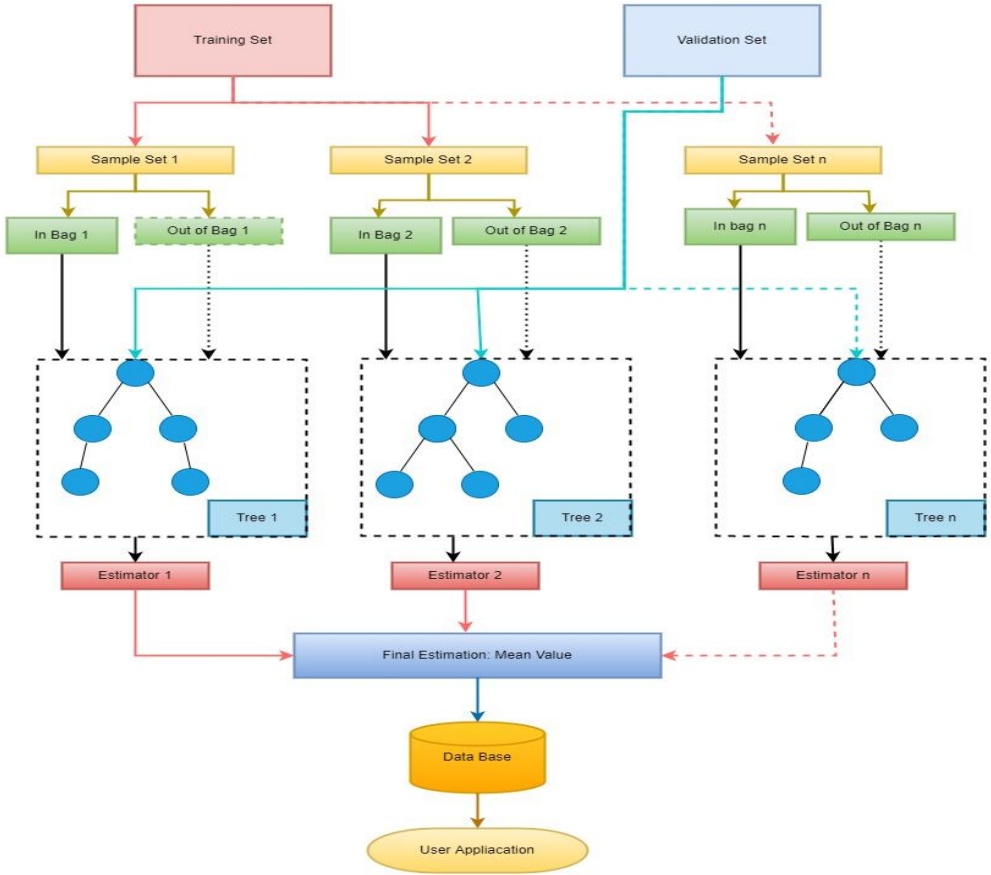
Flowchart



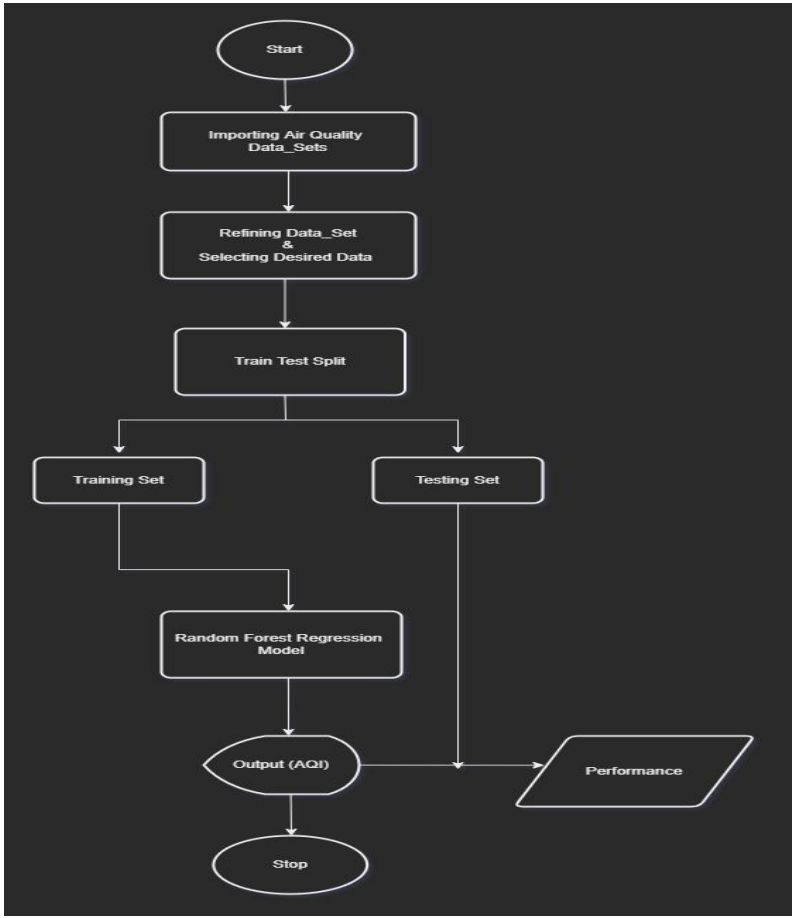
BLOCK DIAGRAM OF PROPOSED ARCHITECTURE



DATA FLOW DIAGRAM



PROCESS FLOW

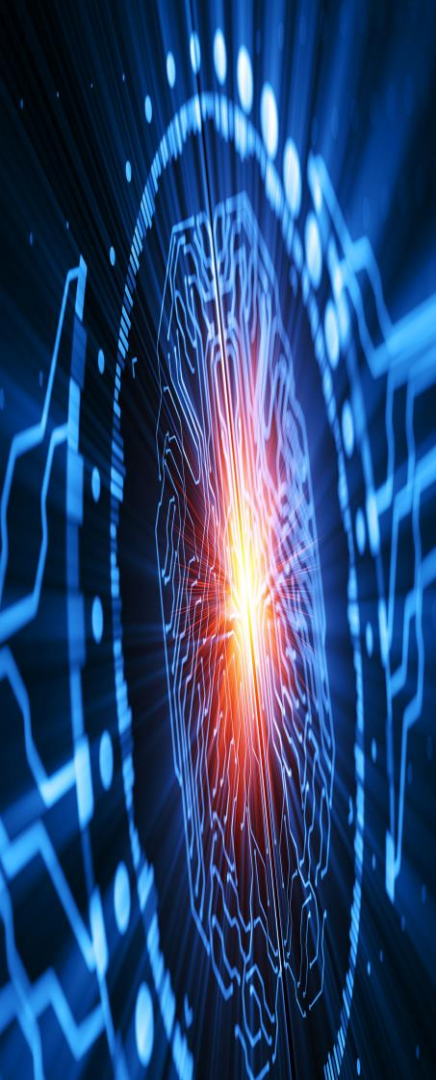


PERFORMANCE STATUS

```
MAE: 10.273708990476212
MSE: 275.9527390853567
RMSE: 16.611825278558545
```

Dependencies / Show stoppers

- Air quality data collected from Kaggle
- IBM Watson Studio Cloud
- Flutter
- Random Forest Algorithm

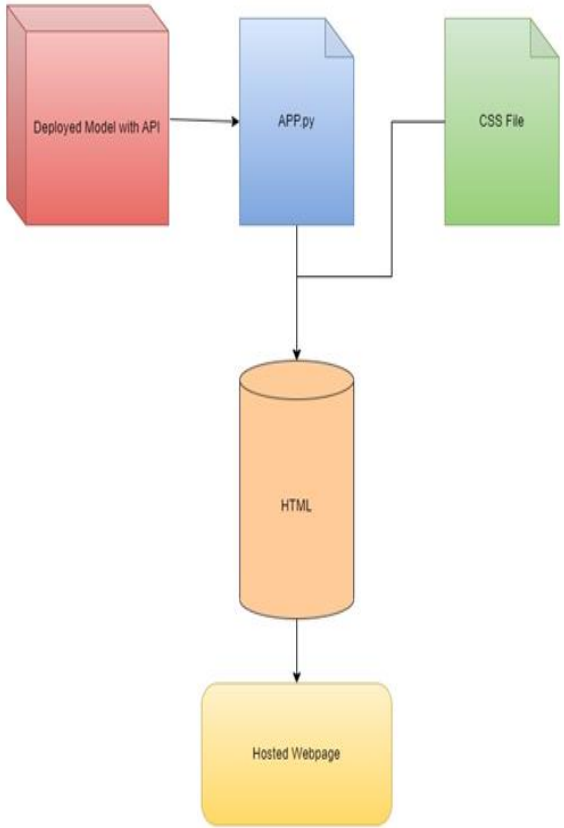


UI / UX DESIGN

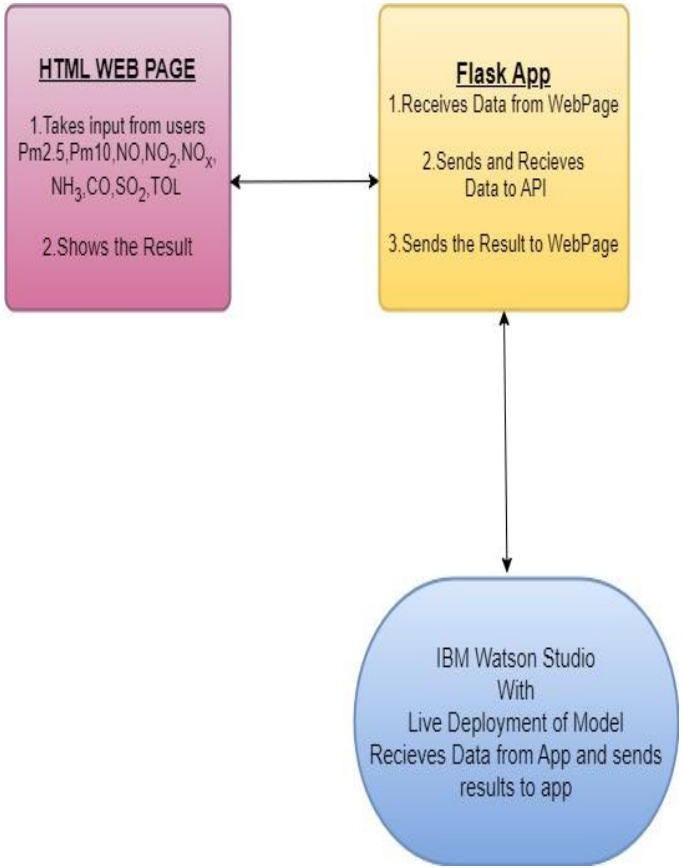
COMPONENTS REQUIRED

- **Programming Language:**
  - HTML
  - CSS
  - Python
- **Framework:**
  - Flask
- **Imported Libraries:**
  - Flask
  - Jsonify
  - Render_template
  - Requests
- **Host:**
  - Local Host
- **Feasibility:**
  - Python (Installed)
  - Browser

FLOWCHART

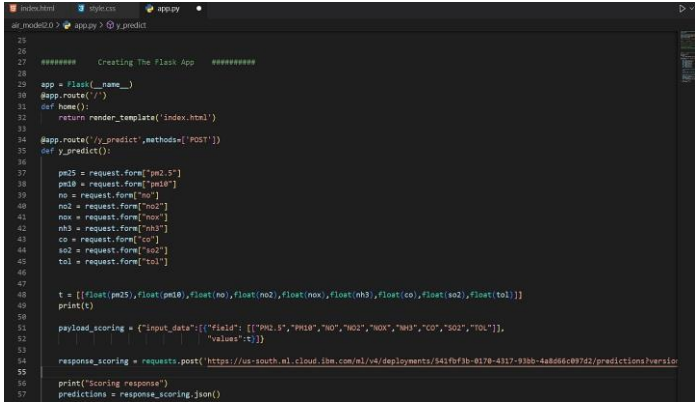


FRAMEWORK : FLASK



1. **Flask:**
   - ✓ Implementation of web frameworks concept
   - ✓ Provides with libraries,tools, modules to develop web appplications
2. **Jsonify:**
   - ✓ Serializes data to JavaScript Object Notation Format (JSON)
   - ✓ Wraps it in a response object with application
2. **Render_template:**
   - ✓ Used to generate output from a template based on Jinja2 engine
3. **Requests:**
   - ✓ Critical for building web applications
   - ✓ Allows us to obtain data sent from client

RESULTS (1)

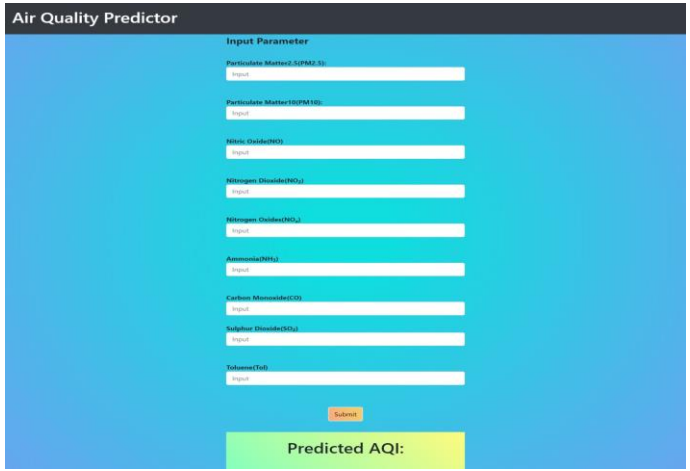


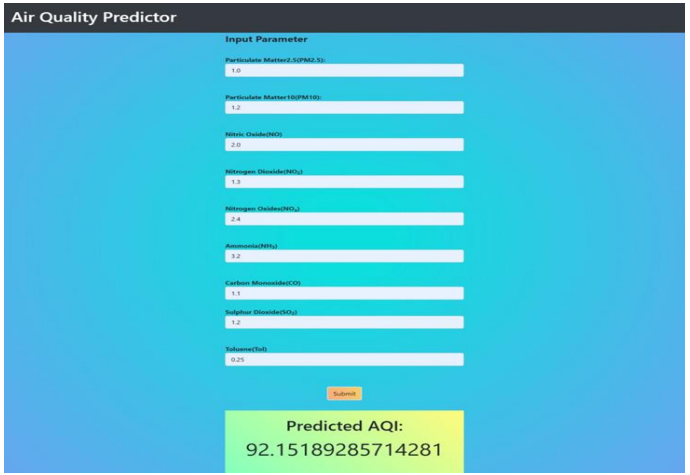
STEP 1: Run the Program

STEP 2: Copy the Link Pointed by the arrow

RESULTS (2)

STEP 3: Paste the link in any Web Browser to get to the User Interface (UI)





STEP 4: Input the values for respective parameters and click on SUBMIT Option to get the prediction

IBM

Smart Internz