

```

1 import streamlit as st
2 import pickle
3
4
5 with open(r"C:\Users\VYSHNAVI\Desktop\internship\projectrf.pkl", 'rb') as file:
6     rf_regressor = pickle.load(file)
7
8
9 st.set_page_config(page_title="🌍 AQI Prediction", layout="wide")
10
11 st.markdown(
12     """
13     <style>
14         .main { background-color: #F5F7FA; }
15         h1 { color: #2E86C1; }
16         h2 { color: #117A65; }
17         .stMetric { font-size: 22px; font-weight: bold; }
18     </style>
19     """, unsafe_allow_html=True
20 )
21
22 st.title("🌍 AQI Prediction Project")
23 st.write("🔍 **Enter atmospheric gas concentrations to predict air quality.**")
24
25
26 st.sidebar.title("🔧 Enter Pollutant Levels")
27 SO2 = st.sidebar.number_input("🟡 SO2 (Sulfur Dioxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
28 CO = st.sidebar.number_input("🔴 CO (Carbon Monoxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
29 NO = st.sidebar.number_input("🟠 NO (Nitric Oxide in ppm)", min_value=0.0, value=0.0, format="%.2f")

```

```

26 st.sidebar.title("🌫️ Enter Pollutant Levels")
27 SO2 = st.sidebar.number_input("🟡 SO2 (Sulfur Dioxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
28 CO = st.sidebar.number_input("🔴 CO (Carbon Monoxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
29 NO = st.sidebar.number_input("🟠 NO (Nitric Oxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
30 NO2 = st.sidebar.number_input("🟣 NO2 (Nitrogen Dioxide in ppm)", min_value=0.0, value=0.0, format="%.2f")
31 NOX = st.sidebar.number_input("⬛ NOX (Nitrogen Oxides in ppm)", min_value=0.0, value=0.0, format="%.2f")
32 NH3 = st.sidebar.number_input("🟢 NH3 (Ammonia in ppm)", min_value=0.0, value=0.0, format="%.2f")
33 O3 = st.sidebar.number_input("💙 O3 (Ozone in ppm)", min_value=0.0, value=0.0, format="%.2f")
34 WS = st.sidebar.number_input("💨 WS (Wind Speed in m/s)", min_value=0.0, value=0.0, format="%.2f")
35 WD = st.sidebar.number_input("🌪️ WD (Wind Direction in degrees)", min_value=0.0, value=0.0, format="%.2f")
36 RH = st.sidebar.number_input("💧 RH (Relative Humidity in %)", min_value=0.0, value=0.0, format="%.2f")
37 SR = st.sidebar.number_input("☀️ SR (Solar Radiation in W/m²)", min_value=0.0, value=0.0, format="%.2f")
38 TC = st.sidebar.number_input("🌡️ TC (Temperature in °C)", min_value=0.0, value=0.0, format="%.2f")
39
40
41 if st.sidebar.button("🔮 Predict AQI"):
42     input_data = [[SO2, CO, NO, NO2, NOX, NH3, O3, WS, WD, RH, SR, TC]]
43     prediction = rf_regressor.predict(input_data)
44
45     st.subheader("📄 Predicted AQI")
46     st.metric(label="AQI Score", value=round(prediction, 2))
47
48
49     if prediction <= 50:
50         category = "✅ Good Air Quality 🌿"
51         color = "green"
52         bar_level = 0.2

```

```
49 if prediction <= 50:
50     category = "✅ Good Air Quality 🌿"
51     color = "green"
52     bar_level = 0.2
53 elif prediction <= 100:
54     category = "🟡 Moderate Air Quality 🧐"
55     color = "blue"
56     bar_level = 0.4
57 elif prediction <= 150:
58     category = "🟠 Unhealthy for Sensitive Groups 🤔"
59     color = "orange"
60     bar_level = 0.6
61 elif prediction <= 200:
62     category = "🔴 Unhealthy Air Quality 🚨"
63     color = "red"
64     bar_level = 0.8
65 else:
66     category = "💀 Hazardous Air Quality ⚠️"
67     color = "darkred"
68     bar_level = 1.0
69
70
71 st.markdown(f"<h3 style='color:{color};'>{category}</h3>", unsafe_allow_html=True)
72 st.progress(bar_level)
73
74 st.warning("🚨 **Health Advisory**: Take precautions based on AQI level.")
75
```