```
import streamlit as st
import requests


api_key="include your api key here"

st.title("Weather and Air Quality Web App")
st.header("Streamlit and AirVisual API")


@st.cache_data
def map_creator(latitude,longitude):
    from streamlit_folium import folium_static
    import folium

    # center on the station
    m = folium.Map(location=[latitude, longitude], zoom_start=10)

    # add marker for the station
    folium.Marker([latitude, longitude], popup="Station",
tooltip="Station").add_to(m)

    # call to render Folium map in Streamlit
    folium_static(m)

@st.cache_data
def generate_list_of_countries():
```

```
    countries_url =
f"https://api.airvisual.com/v2/countries?key={api_key}"

    countries_dict = requests.get(countries_url).json()

    # st.write(countries_dict)

    return countries_dict


@st.cache_data

def generate_list_of_states(country_selected):

    states_url =
f"https://api.airvisual.com/v2/states?country={country_selected}&key=
{api_key}"

    states_dict = requests.get(states_url).json()

    # st.write(states_dict)

    return states_dict


@st.cache_data

def generate_list_of_cities(state_selected,country_selected):

    cities_url =
f"https://api.airvisual.com/v2/cities?state={state_selected}&country=
{country_selected}&key={api_key}"

    cities_dict = requests.get(cities_url).json()

    # st.write(cities_dict)

    return cities_dict


#TODO: Include a select box for the options: ["By City, State, and
Country","By Nearest City (IP Address)","By Latitude and Longitude"]

# and save its selected option in a variable called category


if category == "By City, State, and Country":
```

```
   countries_dict=generate_list_of_countries()
   if countries_dict["status"] == "success":
       countries_list=[]
       for i in countries_dict["data"]:
           countries_list.append(i["country"])
       countries_list.insert(0,"")

       country_selected = st.selectbox("Select a country", options=
                                       countries_list)
       if country_selected:
           # TODO: Generate the list of states, and add a select box
for the user to choose the state

               if state_selected:

                   # TODO: Generate the list of cities, and add a
select box for the user to choose the city

                       if city_selected:
                           aqi_data_url =
f"https://api.airvisual.com/v2/city?city={city_selected}&state={state
_selected}&country={country_selected}&key={api_key}"
                           aqi_data_dict =
requests.get(aqi_data_url).json()

                           if aqi_data_dict["status"] == "success":
                               # TODO: Display the weather and air
quality data as shown in the video and description of the assignment
                           else:
```

```
                                st.warning("No data available for this
location.")


                    else:

                        st.warning("No stations available, please
select another state.")

            else:

                st.warning("No stations available, please select
another country.")

    else:

        st.error("Too many requests. Wait for a few minutes before
your next API call.")


elif category == "By Nearest City (IP Address)":

    url = f"https://api.airvisual.com/v2/nearest_city?key={api_key}"

    aqi_data_dict = requests.get(url).json()


    if aqi_data_dict["status"] == "success":

    # TODO: Display the weather and air quality data as shown in the
video and description of the assignment


    else:

        st.warning("No data available for this location.")


elif category == "By Latitude and Longitude":

    # TODO: Add two text input boxes for the user to enter the
latitude and longitude information


    if latitude and longitude:
```

```
        url =
f"https://api.airvisual.com/v2/nearest_city?lat={latitude}&lon={longi
tude}&key={api_key}"

        aqi_data_dict = requests.get(url).json()


        if aqi_data_dict["status"] == "success":

        # TODO: Display the weather and air quality data as shown in
the video and description of the assignment


        else:

            st.warning("No data available for this location.")
```