```python
import sys
import requests
from PyQt5.QtWidgets import (
    QApplication,
    QMainWindow,
    QLabel,
    QLineEdit,
    QPushButton,
    QVBoxLayout,
    QWidget
)

from PyQt5.QtGui import QFont


class WeatherApp(QMainWindow):
    def_init_(self, api_key):
        super()._init_()


        # Initialize the WeatherApp with the provided API key

        self.api_key = api_key


        # Set the window properties
```

```python
        self.setWindowTitle("Weather App")
        self.setGeometry(200, 200, 350, 250)

        # Create a vertical layout to organize the widgets

        layout = QVBoxLayout()



        # Create a font for labels and input fields

        font = QFont("Arial", 12)



        # Create and configure the location input field
        self.location_label = QLabel("Enter Location:", self)
        self.location_label.setFont(font)
        layout.addWidget(self.location_label)



        # Create an input field for entering the location
        self.location_input = QLineEdit(self)
        self.location_input.setFont(font)
        layout.addWidget(self.location_input)



        # Create labels for displaying weather information
        self.result_label = QLabel("", self)
        self.result_label.setFont(font)
        layout.addWidget(self.result_label)



        self.humidity_label = QLabel("", self)
        self.humidity_label.setFont(font)
        layout.addWidget(self.humidity_label)



        self.wind_label = QLabel("", self)
```

```python
        self.wind_label.setFont(font)
        layout.addWidget(self.wind_label)


        self.visibility_label = QLabel("", self)
        self.visibility_label.setFont(font)
        layout.addWidget(self.visibility_label)



        # Create a button to trigger weather data retrieval
        self.search_button = QPushButton("Search", self)
        self.search_button.setFont(font)
        self.search_button.clicked.connect(self.get_weather)
        layout.addWidget(self.search_button)


        # Create a central widget and set the layout for the main window

        central_widget = QWidget(self)
        central_widget.setLayout(layout)
        self.setCentralWidget(central_widget)


    def get_weather(self):
        # Retrieve the location entered by the user

        location = self.location_input.text()



        # Construct the API URL for weather data retrieval

        api_url                                                    =
f"http://api.worldweatheronline.com/premium/v1/weather.ashx?key={self.api_key}&q={l
ocation}&format=json&num_of_days=1"
```

 # Send a request to the weather API and get the response

```python
response = requests.get(api_url)
data = response.json()
```

 # Extract and display weather information if available

```python
if "data" in data and "current_condition" in data["data"]:
    weather_data = data["data"]["current_condition"][0]
    temperature = weather_data["temp_C"]
    description = weather_data["weatherDesc"][0]["value"]
    humidity = weather_data["humidity"]
    wind_speed = weather_data["windspeedKmph"]
    visibility = weather_data["visibility"]
```

```python
                result_text = f"Weather: {description}, Temperature:
                {temperature}°C"humidity_text = f"Humidity: {humidity}%"
                wind_text = f"Wind Speed: {wind_speed}
                km/h"visibility_text = f"Visibility:
                {visibility} km"
            else:
                result_text = "Weather information not available for this location"
                humidity_text = ""
                wind_text = ""
                visibility_text = ""


                # Update the labels with the retrieved weather
                informationself.result_label.setText(result_text)
                self.humidity_label.setText(humidity_text)
                self.wind_label.setText(wind_text)
                self.visibility_label.setText(visibility_text)


if __name__ == "__main__":
    # Provided API key for accessing weather data
    api_key = "274a6cd71d1945e3a0c62114232410"


    # Create a PyQt application, the WeatherApp instance, and show the main
    window
    app = QApplication(sys.argv)
    weather_app =
    WeatherApp(api_key)
```

```python
weather_app.show()
```

Run the application and event loop

```python
sys.exit(app.exec_())
```