```
import tkinter as tk

from tkinter import font as tkfont, messagebox

import requests



# Replace with your actual OpenWeatherMap API key

API_KEY = "YOUR_API_KEY_HERE"

API_URL = "https://api.openweathermap.org/data/2.5/weather"



class WeatherApp(tk.Tk):

  def __init__(self):

    super().__init__()

    self.title("Weather App")

    self.geometry("480x360")

    self.configure(bg="#ffffff")

    self.resizable(False, False)



    # Fonts

    self.title_font = tkfont.Font(family="Poppins", size=48, weight="bold") # Big bold headline
```

```
    self.headline_font = tkfont.Font(family="Poppins", size=24, weight="bold")

    self.body_font = tkfont.Font(family="Poppins", size=16)

    self.subtext_font = tkfont.Font(family="Poppins", size=14, slant="italic")



    # Header

    header = tk.Label(self, text="Weather App", font=self.title_font, fg="#111827", bg="#ffffff", pady=24)

    header.pack()



    # Input frame

    input_frame = tk.Frame(self, bg="#ffffff", pady=10)

    input_frame.pack(fill="x", padx=40)



    self.city_var = tk.StringVar()

    city_entry = tk.Entry(input_frame, textvariable=self.city_var, font=self.body_font, width=20, bd=2, relief="groove")

    city_entry.pack(side="left", padx=(0, 10))

    city_entry.bind("<Return>", lambda event: self.fetch_weather())
```

```
search_button = tk.Button(
    input_frame,
    text="Search",
    font=self.body_font,
    bg="#111827",
    fg="#ffffff",
    activebackground="#374151",
    activeforeground="#d1d5db",
    bd=0,
    padx=15,
    pady=6,
    command=self.fetch_weather,
    cursor="hand2",
)
search_button.pack(side="left")

# Weather info card
```

```
    self.card = tk.Frame(self, bg="#f9fafb", padx=30, pady=30, bd=1, relief="solid",
highlightbackground="#e5e7eb", highlightcolor="#e5e7eb")

    self.card.pack(padx=40, pady=20, fill="both", expand=True)



    self.location_label = tk.Label(self.card, font=self.headline_font, fg="#111827", bg="#f9fafb")

    self.location_label.pack(anchor="w")



    self.temp_label = tk.Label(self.card, font=self.title_font, fg="#111827", bg="#f9fafb")

    self.temp_label.pack(anchor="w", pady=(5, 0))



    self.weather_label = tk.Label(self.card, font=self.body_font, fg="#6b7280", bg="#f9fafb")

    self.weather_label.pack(anchor="w", pady=(2, 10))



    self.details_label = tk.Label(self.card, font=self.body_font, fg="#6b7280", bg="#f9fafb",
justify="left")

    self.details_label.pack(anchor="w")



  def fetch_weather(self):

    city = self.city_var.get().strip()
```

```
if not city:
    messagebox.showwarning("Input Error", "Please enter a city name.")
    return

params = {
    "q": city,
    "appid": API_KEY,
    "units": "metric",
}

try:
    resp = requests.get(API_URL, params=params, timeout=10)
    resp.raise_for_status()
    data = resp.json()
    self.update_weather(data)
except requests.exceptions.HTTPError as e:
    if resp.status_code == 404:
        messagebox.showerror("Error", f"City '{city}' not found.")
```

```
        else:
            messagebox.showerror("Error", f"API error: {e}")
    except requests.exceptions.RequestException as e:
        messagebox.showerror("Error", f"Network error: {e}")

def update_weather(self, data):
    name = data.get("name", "Unknown")
    sys = data.get("sys", {})
    country = sys.get("country", "")
    main = data.get("main", {})
    weather = data.get("weather", [{}])[0]
    wind = data.get("wind", {})

    temp = main.get("temp", "N/A")
    desc = weather.get("description", "N/A").capitalize()
    humidity = main.get("humidity", "N/A")
    wind_speed = wind.get("speed", "N/A")
```

```
        self.location_label.config(text=f"{name}, {country}")
        self.temp_label.config(text=f"{temp}°C")
        self.weather_label.config(text=desc)
        self.details_label.config(text=f"Humidity: {humidity}%\nWind Speed: {wind_speed} m/s")

if __name__ == "__main__":
    app = WeatherApp()
    app.mainloop()
```