```
 1 import requests
 2
 3 # Function to fetch weather information based on zip
   code
 4 def find_Ziptemp(api_key, base_url, geo_url_zip,
   zip_Code , temp_unit):
 5     zip_Url = geo_url_zip + "zip=" + zip_Code + ",us&
   appid=" + api_key
 6     try:
 7         response = requests.get(zip_Url)
 8         response.raise_for_status()
 9         formatted_response = response.json()
10         if formatted_response.get("cod") != "404":
11             latitude = str(formatted_response["lat"])
12             longitude = str(formatted_response["lon"
   ])
13             z_url = base_url + "lat=" + latitude + "&
   lon=" + longitude + "&appid=" + api_key + temp_unit
14             response_2 = requests.get(z_url)
15             response_2.raise_for_status()
16             formatted_response_2 = response_2.json()
17             if formatted_response_2.get("cod") != "
   404":
18                 display_weather(formatted_response_2)
19             else:
20                 print("City Not Found, Please try
   again")
21         else:
22             print("City Not Found, Please try again")
23     except requests.exceptions.RequestException as
   err:
24         print("Error occurred:", err)
25
26 # Function to fetch weather information based on city
    name and state
27 def find_Citytemp(api_key, base_url, geo_url_city,
   city_Name, state_Name, temp_unit):
28     city_Url = geo_url_city + "q=" + city_Name + ","
    + state_Name + ",us&appid=" + api_key
29     try:
30         response = requests.get(city_Url)
```

```
        response.raise_for_status()
        formatted_response = response.json()
        if formatted_response:
            latitude = str(formatted_response[0]["lat
"])
            longitude = str(formatted_response[0]["
lon"])
            c_url = base_url + "lat=" + latitude + "&
lon=" + longitude + "&appid=" + api_key + temp_unit
            response_2 = requests.get(c_url)
            response_2.raise_for_status()
            formatted_response_2 = response_2.json()
            if formatted_response_2.get("cod") != "
404":
                display_weather(formatted_response_2)
            else:
                print("City Not Found, Please try
again")
        else:
            print("City Not Found, Please try again")
    except requests.exceptions.RequestException as
err:
        print("Error occurred:", err)

# Function to display weather information
def display_weather(formatted_response):
    main_list = formatted_response["main"]
    current_Temp = main_list["temp"]
    high_Temp = main_list["temp_max"]
    low_Temp = main_list["temp_min"]
    current_pressure = main_list["pressure"]
    current_Humidity = main_list["humidity"]
    clouds_list = formatted_response["weather"]
    weather_description = clouds_list[0]["description
"]
    print("\nTemperature = " +
            str(current_Temp) + " degrees" +
        "\nTemperature High = " +
            str(high_Temp) + " degrees" +
        "\nTemperature Low = " +
            str(low_Temp) + " degrees" +
```

```
65          "\nPressure = " +
66              str(current_pressure) + " hPa" +
67          "\nHumidity = " +
68              str(current_Humidity) + " %" +
69          "\nCloud Cover = " +
70              str(weather_description))
71
72  def main():
73      api_key = "ENTER API KEY HERE"
74      base_url = "https://api.openweathermap.org/data/
    2.5/weather?"
75      geo_url_city = "http://api.openweathermap.org/
    geo/1.0/direct?"
76      geo_url_zip = "http://api.openweathermap.org/geo
    /1.0/zip?"
77
78      while True:
79          user_Need= input("Would you like to lookup
    weather by City Name or Zip Code? Enter 1 for City,
    2 for Zip code: ")
80          if user_Need == '1':
81              city_Name = input("Please enter City
    Name: ")
82              state_Name = input("Please enter State
    Abbreviation: ")
83              temp_Need= input("Would you like to view
     temps in Fahrenheit, Celsius, or Kelvin?"
84                           "\nEnter 'F' for Fahrenheit, '
    C' for Celsius, 'K' for Kelvin: ").upper()
85              if temp_Need not in ['F', 'C', 'K']:
86                  print("Error, please restart and
    press F, C, or K for temperature")
87                  break
88              temp_unit = "&units=imperial" if
    temp_Need == 'F' else "&units=metric" if temp_Need
     == 'C' else ''
89              find_Citytemp(api_key, base_url,
    geo_url_city, city_Name, state_Name, temp_unit)
90          elif user_Need == '2':
91              zip_Code = input("Please enter Zip Code
    : ")
```

```
            temp_Need= input("Would you like to view temps in Fahrenheit, Celsius, or Kelvin?"
                             "\nEnter 'F' for Fahrenheit, 'C' for Celsius, 'K' for Kelvin: ").upper()
            if temp_Need not in ['F', 'C', 'K']:
                print("Error, please restart and press F, C, or K for temperature")
                break
            temp_unit = "&units=imperial" if temp_Need == 'F' else "&units=metric" if temp_Need == 'C' else ''
            find_Ziptemp(api_key, base_url, geo_url_zip, zip_Code, temp_unit)
        else:
            print("Error, please restart and press 1 or 2 for city or zip")
            break

        repeat = input("Would you like to perform another weather lookup? Y/N").upper()
        if repeat != 'Y':
            print("\nThank you!")
            break

if __name__ == "__main__":
    main()
```