Weather App

April 7, 2024

```python
[1]: import requests
     import json
     import time

     print('''Welcome to Weather Channel!!!\nAccess Current Weather Data & forecast␣
      ↪for any US cities!!\n''')
```

```
Welcome to Weather Channel!!!
Access Current Weather Data & forecast for any US cities!!
```

```python
[2]: def main():
         """Main function takes the user input as zip code or city/country name to␣
      ↪receive current/future weather forecast"""
         url = 'https://api.openweathermap.org/data/2.5/weather'
         url_ext = 'https://api.openweathermap.org/data/2.5/forecast'
         location = input('\nPlease enter the Zip Code or City, Country: ')
         while True:
             try:
                 weather_current(location, url)
                 weather_extended(location, url_ext)
                 print('')
                 more_weather()
                 break
             except LookupError:
                 print('')
                 more_weather()
                 break
```

```python
[3]: def weather_current(location, url):
         """Makes a GET request to the url for current weather, verifies connection␣
      ↪is made, parses and displays the data"""
         if location.isdigit() is True:
             query_params = {'zip': location, 'APPID':␣
      ↪'e0658f792164bea0f30488a83ec7f9c9'}
         else:
```

```python
        query_params = {'q': location, 'APPID':
 ↪'e0658f792164bea0f30488a83ec7f9c9'}
    response = requests.get(url, params=query_params, timeout=(5, 14))
    try_web(response, location)
    if response.status_code == 200:
        print('Connected....Location Found')
    current_parsed = json.loads(response.text)
    current_formatted(current_parsed)
```

```python
[11]: def weather_extended(location, url_ext):
    """Makes a GET request to the url for extended forecast, parses and
 ↪displays the data"""
    if location.isdigit() is True:
        query_params = {'zip': location, 'cnt': 16, 'APPID':
 ↪'af3008006e6d4caf44f8d3b922cdf469'}
    else:
        query_params = {'q': location, 'cnt': 16, 'APPID':
 ↪'af3008006e6d4caf44f8d3b922cdf469'}
    response = requests.get(url_ext, params=query_params, timeout=(5, 14))
    try_web(response, location)
    ext_parsed = json.loads(response.text)
    ext_formatted(ext_parsed)


def convert_temp(temp):
    """Converts Kelvin temperatures to Fahrenheit and Celsius"""
    f_degree = round((((temp - 273.15)*9)/5)+32)
    c_degree = round(temp - 273.15)
    return f'{f_degree}{chr(176)}F / {c_degree}{chr(176)}C'
```

```python
[12]: def try_web(response, location):
    """Try Except block to test the request was successful, additionally
 ↪checking if the city or
    zip code entered is valid by using 404 status code"""
    try:
        response.raise_for_status()
    except requests.HTTPError as error0:
        if response.status_code == 404:
            if location.isdigit() is True:
                print(f"The zip code entered '{location}' was not found or is
 ↪not valid.")
            else:
                if location.__contains__(','):
                    print(f"The city entered '{location[0:-2].title() +
 ↪location[-2:].upper()}' was not found.")
                else:
```

```python
                    print(f"The city entered '{location.title()}' was not found.
↪")
            else:
                print('Even we do not have access to single digit zip codes.')
                print(f'{error0}')
        except requests.ConnectionError as error1:
            print('Error Connecting')
            print(error1)
        except requests.Timeout as error2:
            print('Timeout Error')
            print(error2)
        except requests.RequestException as error3:
            print('Something Else Went Wrong')
            print(error3)
```

```python
[13]: def current_formatted(parsed):
          """Decodes the JSON data, formats the time variables to match proper time␣
      ↪zones, then formats the printable
          output of the current weather"""
          city = str(json.dumps(parsed['name'])).replace('"', '')
          country = str(json.dumps(parsed['sys']['country'])).replace('"', '')
          timezone = int(json.dumps(parsed['timezone']))
          epoch_time = int(json.dumps(parsed['dt']))
          true_time = epoch_time + timezone
          current_time = time.strftime("%A, %b %d, %Y %I:%M %p (local time)", time.
      ↪gmtime(true_time))
          temp = float(json.dumps(parsed['main']['temp']))
          conditions = str(json.dumps(parsed['weather'][0]['description'])).
      ↪replace('"', '').title()
          print(f'Weather Report for {city}, {country} on {current_time}:\n'
                f'Current Temperature {convert_temp(temp)}\n'
                f'Current Conditions: {conditions}\n')
```

```python
[14]: def ext_formatted(parsed):
          """Decodes the JSON data, formats the time variables to match time to the␣
      ↪time zones, then formats the printable
          output of the extended forecast"""
          print(f"{'36 Hour Forecast':30}{'Temperature':22}{'Conditions'}")
          # For loop to pull the data for every six (6) hours, approximate 36 hour␣
      ↪forecast data return
          for i in range(1, 15, 2):
              epoch_time = int(json.dumps(parsed['list'][i]['dt']))
              timezone = int(json.dumps(parsed['city']['timezone']))
              true_time = epoch_time + timezone
              future_time = time.strftime("%a, %b %d %I:%M %p", time.
      ↪gmtime(true_time))
```

```python
        temp = float(json.dumps(parsed['list'][i]['main']['temp']))
        conditions = str(json.
    dumps(parsed['list'][i]['weather'][0]['description'])).replace('"', '').
    title()
        print(f'{future_time:30}{convert_temp(temp):22}{conditions}')


def more_weather():
    """Allows the user to look up another location or exit the program"""
    option = str(input('Would you like to enter another location, Yes or No?
    ')).lower().strip()
    # while loop for a yes selection or to exit the program (and to catch input
    errors)
    while not (option == 'yes' or option == 'no'):
        option = str(input('You did not enter a valid selection.\n'
                           'Please enter Yes to search another location or No
    to exit: ')).lower().strip()
    if option == 'yes':
        print('')
        main()
    if option == 'no':
        print('Thank you for using our service. Goodbye')
```

```python
[15]: if __name__ == "__main__":
    main()
```

```
Please enter the Zip Code or City, Country: 60173
Connected…Location Found
Weather Report for Schaumburg, US on Sunday, Feb 19, 2023 05:31 PM (local time):
Current Temperature 46°F / 8°C
Current Conditions: Clear Sky

36 Hour Forecast             Temperature           Conditions
Sun, Feb 19 09:00 PM         44°F / 6°C            Scattered Clouds
Mon, Feb 20 03:00 AM         32°F / 0°C            Broken Clouds
Mon, Feb 20 09:00 AM         34°F / 1°C            Scattered Clouds
Mon, Feb 20 03:00 PM         45°F / 7°C            Scattered Clouds
Mon, Feb 20 09:00 PM         40°F / 4°C            Overcast Clouds
Tue, Feb 21 03:00 AM         34°F / 1°C            Overcast Clouds
Tue, Feb 21 09:00 AM         29°F / -2°C           Clear Sky


Would you like to enter another location, Yes or No? no
Thank you for using our service. Goodbye
```

```python
[ ]:
```