

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING		
<b>Program Name:</b> M.Tech/MSC/MCA		<b>Assignment Type:</b> Lab		
<b>Course Coordinator Name</b>		Venkataramana Veeramsetty		
<b>Course Code</b>		<b>Course Title</b>	AI Assisted Problem Solving Using Python	
<b>Year/Sem</b>		<b>Regulation</b>	R25	
<b>Date and Day of Assignment</b>	Week5 - Tuesday	<b>Time(s)</b>		
<b>Duration</b>	2 Hours	<b>Applicable to Batches</b>		
<b>AssignmentNumber:</b> 15.4(Present assignment number)/ <b>24</b> (Total number of assignments)				

<b>Q.No.</b>	<b>Question</b>	<b>Expected Time to complete</b>
1	<p><b>Lab 18 – API Integration: Connecting to External Services with Error Handling</b></p> <p><b>Lab Objectives:</b></p> <ul style="list-style-type: none"> <li>• Learn how to integrate Python programs with external REST APIs.</li> <li>• Understand API request/response handling using requests or similar libraries.</li> <li>• Implement proper error handling for failed API calls (timeouts, invalid responses, rate limits).</li> <li>• Practice extracting and displaying meaningful information from API responses.</li> </ul>	Week5 - Tuesday
	<p><b>Task 1: Connect to a Public API</b></p> <ul style="list-style-type: none"> <li>• <b>Instructions:</b> <ul style="list-style-type: none"> <li>• Use Python (or Node.js/JavaScript) to connect to a public API (e.g., OpenWeatherMap or JSONPlaceholder).</li> <li>• Send a simple GET request to retrieve data.</li> <li>• Display the response in a readable format (pretty JSON).</li> </ul> </li> <li>• <b>Expected Output:</b> <ul style="list-style-type: none"> <li>◦ Program prints API response (e.g., weather data or posts from JSONPlaceholder).</li> <li>◦ Output should be properly formatted JSON (not raw).</li> </ul> </li> </ul>	

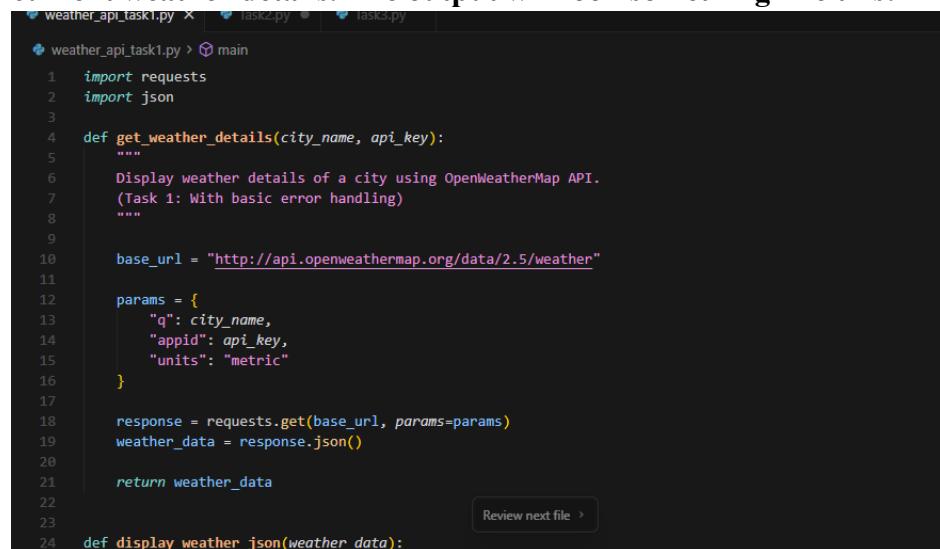
**Prompt: write a python function that display weather details of a city using weather api without error handling. Display weather details as JSON output**

**To use this code:**

1. First, you'll need to sign up for a free API key at Open Weather Map: <https://openweathermap.org/api>
2. Replace 'YOUR\_API\_KEY' in the code with your actual API key
3. Install the required requests package by running:

**pip install requests**

**You can then run the script and enter any city name to get its current weather details. The output will look something like this:**



```
weather_api_task1.py > main
1 import requests
2 import json
3
4 def get_weather_details(city_name, api_key):
5     """
6         Display weather details of a city using OpenWeatherMap API.
7         (Task 1: With basic error handling)
8     """
9
10    base_url = "http://api.openweathermap.org/data/2.5/weather"
11
12    params = {
13        "q": city_name,
14        "appid": api_key,
15        "units": "metric"
16    }
17
18    response = requests.get(base_url, params=params)
19    weather_data = response.json()
20
21    return weather_data
22
23
24 def display_weather_json(weather_data):
```

```
weather_api_task1.py > main
24  def display_weather_json(weather_data):
25      formatted_json = json.dumps(weather_data, indent=2, ensure_ascii=False)
26      print("\n" + "="*60)
27      print("Weather Details (JSON Format):")
28      print("="*60)
29      print(formatted_json)
30      print("="*60 + "\n")
31
32
33 def main():
34     print("Weather API - City Weather Details")
35     print("-" * 60)
36
37     # Replace with Your Active API Key
38     api_key = "5b54945900fa4db64d400472c9daea1"
39
40     city_name = input("Enter the city name: ").strip()
41     print(f"\nFetching weather data for {city_name}...")
42
43     weather_data = get_weather_details(city_name, api_key)
```

```
Problems Output Debug Console Terminal Ports
PS C:\Users\imgop\OneDrive\Desktop\Assig15> & "C:/Program Files/Python314/python.exe" c:/Users/imgop/OneDrive/Desktop/Assig15/Task2.py
● Weather API - City Weather Details (Task 2: With Error Handling)
-----
Enter the city name: Kolkata
Fetching weather data for Kolkata...
=====
Weather Details (JSON Format):
=====
{
  "coord": {
    "lon": 88.3697,
    "lat": 22.5697
  },
  "weather": [
    {
      "id": 721,
      "main": "Haze",
      "description": "haze",
      "icon": "50d"
    }
  ]
}
```

```
        },
      ],
      "base": "stations",
      "main": {
        "temp": 22.97,
        "feels_like": 23.1,
        "temp_min": 22.97,
        "temp_max": 22.97,
        "pressure": 1017,
        "humidity": 68,
        "sea_level": 1017,
        "grnd_level": 1017
      },
      "visibility": 2000,
      "wind": {
        "speed": 2.06,
        "deg": 10
      },
      "clouds": {
        "all": 0
      },
      "dt": 1764218056,
      "sys": {
        "type": 1,
        "id": 9114,
        "country": "IN",
        "sunrise": 1764203239,
        "sunset": 1764242475
      }
    }
  }
}
```

## Task 2: Add Error Handling for Invalid API Calls

- **Instructions:**

- Modify your code from Task 1 to handle errors.
- Include try/except (Python) or try/catch (JavaScript) blocks.
- Handle cases like:
  - Invalid URL
  - Network timeout
  - Wrong API key (if required)
- Print user-friendly error messages.

- **Expected Output:**

- If the API works, the result is shown as in Task 1.
- If there's an error, output:

**Error: Could not connect to API. Check your API key or network connection.**

**Prompt: write a python function that display weather details of a city using weather api with error handling. Display weather details as JSON output**

**To use this code:**

1. First, you'll need to sign up for a free API key at Open

## Weather Map: <https://openweathermap.org/api>

2. Replace 'YOUR\_API\_KEY' in the code with your actual API key
3. Install the required requests package by running:

```
Task2.py > main
1 import requests
2 import json
3
4 def get_weather_details(city_name, api_key):
5     """
6         Display weather details of a city using OpenWeatherMap API.
7         (Task 2: With comprehensive error handling)
8
9     Args:
10        city_name (str): Name of the city to get weather for
11        api_key (str): OpenWeatherMap API key
12
13    Returns:
14        dict: Weather data as JSON, or None if error occurs
15    """
16
17    # OpenWeatherMap API endpoint
18    base_url = "http://api.openweathermap.org/data/2.5/weather"
19
20    # Parameters for the API request
21    params = {
22        "q": city_name,
23        "appid": api_key,
24        "units": "metric" # Use metric units (Celsius)
25    }
26
27    try:
28        Review next file >
```

```
task2.py > main
1 C:\Users\imgop\OneDrive\Desktop\Assig15\Task2.py >i_key):
2     try:
3         # Send GET request to the API with timeout
4         response = requests.get(base_url, params=params, timeout=10)
5
6         # Check if the request was successful (status code 200)
7         response.raise_for_status()
8
9         # Parse JSON response
10        weather_data = response.json()
11
12        # Check if API returned an error in the JSON response
13        if weather_data.get("cod") and str(weather_data.get("cod")) != "200":
14            # API returned error code (e.g., 401 for invalid API key, 404 for city not found)
15            if weather_data.get("cod") == 401:
16                print("Error: Could not connect to API. Check your API key or network connection.")
17            else:
18                print("Error: Could not connect to API. Check your API key or network connection.")
19            return None
20
21        # Return the weather data
22        return weather_data
23
24    except requests.exceptions.Timeout:
25        print("Error: Could not connect to API. Review next file > key or network connection.")
26        return None
```

```

Problems Output Debug Console Terminal Ports
PS C:\Users\imgop\OneDrive\Desktop\Assig15> & "C:/Program Files/Python314/python.exe" c:/Users/imgop/OneDrive/Desktop/
Assig15/Task2.py
Weather API - City Weather Details (Task 2: With Error Handling)
-----
Enter the city name: Kolkata
Fetching weather data for Kolkata...
Error: Could not connect to API. Check your API key or network connection.
PS C:\Users\imgop\OneDrive\Desktop\Assig15>

Enter the city name: Kolkata
Fetching weather data for Kolkata...
Enter the city name: Kolkata
Enter the city name: Kolkata
Fetching weather data for Kolkata...
Enter the city name: Kolkata
Enter the city name: Kolkata

```

### Task 3: Extract and Display Specific Data

- Instructions:**

1. From the API response (e.g., weather API), extract specific fields (temperature, humidity, description).
2. Display them in a user-friendly format (not raw JSON).

- Expected Output:**

- City: London
- Temperature: 18°C
- Humidity: 60%
- Weather: Clear sky

```

Task3.py > main
1 import requests
2 import json
3
4 def get_weather_details(city_name, api_key):
5     """
6         Retrieve weather details of a city using OpenWeatherMap API.
7         (Task 3: Extract and Display Specific Data)
8
9     Args:
10         city_name (str): Name of the city to get weather for
11         api_key (str): OpenWeatherMap API key
12
13     Returns:
14         dict: Weather data as JSON, or None if error occurs
15     """
16
17     # OpenWeatherMap API endpoint
18     base_url = "http://api.openweathermap.org/data/2.5/weather"
19
20     # Parameters for the API request
21     params = {
22         "q": city_name,
23         "appid": api_key,
24         "units": "metric" # Use metric units (Celsius)
25     }

```

```
def get_weather_details(city_name, api_key):
    """
    Send GET request to the API with timeout
    response = requests.get(base_url, params=params, timeout=10)

    Check if the request was successful (status code 200)
    response.raise_for_status()

    Parse JSON response
    weather_data = response.json()

    Check if API returned an error in the JSON response
    if weather_data.get("cod") and str(weather_data.get("cod")) != "200":
        # API returned error code (e.g., 401 for invalid API key, 404 for city not found)
        if weather_data.get("cod") == 401:
            print("Error: Could not connect to API. Check your API key or network connection.")
        else:
            print("Error: Could not connect to API. Check your API key or network connection.")
    return None

    Return the weather data
    return weather_data
except requests.exceptions.Timeout:
```

[Review next file >](#)

Problems Output Debug Console Terminal Ports

Enter the city name: Warangal

Fetching weather data for Warangal...  
Error: Could not connect to API. Check your API key or network connection.

PS C:\Users\imgop\OneDrive\Desktop\Assig15> & "C:/Program Files/Python314/python.exe" c:/Users/imgop/Desktop/Assig15/Task3.py  
Weather API - Extract and Display Specific Data (Task 3)

-----

Enter the city name: Warangal

Fetching weather data for Warangal...

City: Warangal  
Temperature: 24°C  
Humidity: 62%  
Weather: Scattered Clouds

Humidity: 62%  
Weather: Scattered Clouds

PS C:\Users\imgop\OneDrive\Desktop\Assig15> []

#### Task 4: Build a Function with Parameters

- **Instructions:**

- Write a function that accepts a parameter (e.g., city name for weather API).
- The function should call the API dynamically based on user input.
- Include error handling if the city is invalid.

- **Expected Output:**

- Input: "New York"
- Output:
- City: New York
- Temperature: 22°C
- Humidity: 55%
- Weather: Few clouds
- Input: "xyz123"
- Output:

**Error: City not found. Please enter a valid city.**

```
Task4.py > ...
1 import requests
2 import json
3
4 def get_weather_by_city(city_name):
5     """
6         Get weather details for a city using OpenWeatherMap API.
7         (Task 4: Build a Function with Parameters)
8
9     This function accepts a city name as a parameter and dynamically calls the API.
10
11    Args:
12        city_name (str): Name of the city to get weather for
13
14    Returns:
15        dict: Weather data as JSON if successful, None if error occurs
16    """
17
18    # API key - Replace with your actual OpenWeatherMap API key
19    # Get your free API key at: https://openweathermap.org/api
20    api_key = "5b54945900fa4db64d400472c9daae1"
21
22    # OpenWeatherMap API endpoint
23    base_url = "http://api.openweathermap.org/data/2.5/weather"
24
25    # Parameters for the API request
```

```
Task4.py > ...
4  C:\Users\imgop\OneDrive\Desktop\Assig15\Task4.py
5  # Parameters for the API request
6  params = {
7      "q": city_name,
8      "appid": api_key,
9      "units": "metric" # Use metric units (Celsius)
10 }
11
12 try:
13     # Send GET request to the API with timeout
14     response = requests.get(base_url, params=params, timeout=10)
15
16     # Parse JSON response
17     weather_data = response.json()
18
19     # Check if API returned an error in the JSON response
20     if weather_data.get("cod") and str(weather_data.get("cod")) != "200":
21         # City not found (404) or invalid API key (401)
22         if weather_data.get("cod") == 404:
23             print("Error: City not found. Please enter a valid city.")
24             return None
25         elif weather_data.get("cod") == 401:
26             print("Error: Could not connect to API. Check your API key or network connection.")
27             return None
28         else:
29             print("Review next file >")
30
31     print("Error: City not found. Please enter a valid city.")
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

Problems Output Debug Console Terminal Ports

Assig15/Task4.py  
Weather API - Function with Parameters (Task 4)

---

Enter the city name: Kolkata

Fetching weather data for Kolkata...

Fetching weather data for Kolkata...

Fetching weather data for Kolkata...

City: Kolkata  
City: Kolkata  
Temperature: 24°C  
Humidity: 57%  
Humidity: 57%  
Weather: Haze

PS C:\Users\imgop\OneDrive\Desktop\Assig15>

**Prompt: write a python function that display weather details of a city using weather api with error handling. Display weather details in user friendly format**

## Task 5: Store API Results Locally

- **Instructions:**

- Extend your function from Task 4.
- Save the extracted API results into a local file (results.json or results.txt).
- Each new request should append results without overwriting old ones.

- **Expected Output:**

- Console still shows formatted output.
- A local file results.json is created/updated with stored responses like:
  - [
  - {"city": "London", "temp": 18, "humidity": 60, "weather": "Clear sky"},
  - {"city": "New York", "temp": 22, "humidity": 55, "weather": "Few clouds"}

]

```
task5.py >....  
1 C:\Users\Imgop\OneDrive\Desktop\Assig15\Task5.py  
2 import json  
3 import os  
4 from datetime import datetime  
5  
6 def get_weather_by_city(city_name):  
7     """  
8         Get weather details for a city using OpenWeatherMap API.  
9         (Task 5: Store API Results Locally - Extended from Task 4)  
10  
11     This function accepts a city name as a parameter and dynamically calls the API.  
12  
13     Args:  
14         city_name (str): Name of the city to get weather for  
15  
16     Returns:  
17         dict: Weather data as JSON if successful, None if error occurs  
18     """  
19     # API key - Replace with your actual OpenWeatherMap API key  
20     # Get your free API key at: https://openweathermap.org/api  
21     api_key = "5b54945900fa4db64d400472c9daae1"  
22  
23     # OpenWeatherMap API endpoint  
24     base_url = "http://api.openweathermap.org/data/2.5/weather"  
25  
26     # Parameters for the API request
```

```

def get_weather_by_city(city_name):
    # Parameters for the API request
    params = {
        "q": city_name,
        "appid": api_key,
        "units": "metric" # Use metric units (Celsius)
    }

    try:
        # Send GET request to the API with timeout
        response = requests.get(base_url, params=params, timeout=10)

        # Parse JSON response
        weather_data = response.json()

        # Check if API returned an error in the JSON response
        if weather_data.get("cod") and str(weather_data.get("cod")) != "200":
            # City not found (404) or invalid API key (401)
            if weather_data.get("cod") == 404:
                print("Error: City not found. Please enter a valid city.")
                return None
            elif weather_data.get("cod") == 401:
                print("Error: Could not connect to API. Check your API key or network connection.")
                return None
            else:
                print("Error: City not found. Please enter a valid city.")

    except requests.exceptions.RequestException as e:
        print(f"An error occurred: {e}")

```

```

Problems Output Debug Console Terminal Ports
-----
Enter the city name: Warangal
-----
Fetching weather data for Warangal...
City: Warangal
Temperature: 24°C
Fetching weather data for Warangal...
City: Warangal
Temperature: 24°C
○ Temperature: 24°C
Humidity: 62%
Humidity: 62%
Weather: Scattered Clouds
√ Results saved to results.json and results.txt
PS C:\Users\imgop\OneDrive\Desktop\Assig15>

```

**Prompt: write a python function that display weather details of a city using weather api with error handling. Display weather details as JSON output. Store the weather details in current directory as text file, every run output will append**

#### Deliverables (For All Tasks)

1. AI-generated prompts for code and test case generation.
2. At least 3 assert test cases for each task.
3. AI-generated initial code and execution screenshots.
4. Analysis of whether code passes all tests.
5. Improved final version with inline comments and explanation.
6. Compiled report (Word/PDF) with prompts, test cases, assertions, code, and output.