SE Assignment-2

1. Requirements for the digital-based future energies program initiative:
   a. Integrated platform for managing geographically dispersed energy data.
   b. Real-time monitoring and tracking of assets.
   c. Clear and traceable objectives to ensure transparency in the management of renewable energies.
   d. Automated processes for informed decision making.
   e. Predictive maintenance to reduce downtime and increase efficiency.
   f. Accurate forecasting of weather and market conditions for maximising renewable energy production.
   g. Use of artificial intelligence and machine learning for optimising engineering and construction of new renewable sources.
   h. Installation of sensors and counters to create smart grids.
   i. Interconnection of all assets for centralised management.
   j. Reduction of carbon footprint in operations.
2. Technologies, tools and systems to support the needs:
   a. Integrated platform - Siemens Digital Energy Management System, GE Digital Power Digital Energy Management Platform, Schneider Electric EcoStruxure Energy Management System.
   b. Real-time monitoring and tracking of assets - Honeywell Enraf Terminal Management System, Emerson DeltaV DCS, ABB Ability AssetVista.
   c. Clear and traceable objectives - SAP S/4HANA for Energy and Natural Resources, IBM Maximo Asset Management.
   d. Automated processes - GE Digital Plant Applications, Siemens Digital Plant Framework, ABB Ability System 800xA.
   e. Predictive maintenance - OSIsoft PI System, IBM Maximo Predictive Maintenance Insights, Hitachi Vantara Lumada.
   f. Accurate forecasting of weather and market conditions - IBM Watson IoT, GE Digital Grid Analytics, Schneider Electric EcoStruxure Weather.
   g. Use of artificial intelligence and machine learning - Microsoft Azure Machine Learning, Google Cloud AI Platform, Amazon SageMaker.
   h. Smart grids - Siemens Spectrum Power 7, Schneider Electric EcoStruxure Grid.
   i. Interconnection of all assets - Siemens Mindsphere, GE Predix, ABB Ability.

j. Reduction of carbon footprint - Carbon Lighthouse, Enablon Sustainability Management, Schneider Electric EcoStruxure Resource Advisor.

3. Working API code:

- import the requests library and define the API endpoint and parameters

```
import requests

api_endpoint = 'https://api.openweathermap.org/data/2.5/weather'
api_key = 'your_api_key'
location = 'New York'
params = {
    'q': location,
    'appid': api_key,
    'units': 'metric'
}
```

- we are using the `q` parameter to specify the location, `appid` to specify our API key, and `units` to specify the units we want the data to be returned in.
- We can now make a GET request to the API endpoint using the `requests.get()` method and passing in our API endpoint and parameters:

```
response = requests.get(api_endpoint, params=params)
```

- We can then check the response status code to ensure that the API call was successful:

```
if response.status_code == 200:
    data = response.json()
    print(data)
else:
    print('Error: Failed to retrieve data')
```

- If the status code is 200, we can parse the JSON data returned by the API call and perform some basic analysis:

```
if response.status_code == 200:
    data = response.json()
    temperature = data['main']['temp']
    description = data['weather'][0]['description']
    wind_speed = data['wind']['speed']

    print(f'Current weather in {location}:')
    print(f'Temperature: {temperature}°C')
    print(f'Description: {description}')
    print(f'Wind Speed: {wind_speed} m/s')
else:
    print('Error: Failed to retrieve data')
```

- This code will retrieve weather data for the specified location and print out the temperature, description, and wind speed.