

ASSINGMENT-2

Name: R.LOHITH VARMA

Roll No: 21bcs095

1. List of requirements:

Prompt provided in chatgpt :

Generate a list of various requirements(scope) for the above program initiative that can be used for developing a suitable technology oriented digital solution.

ChatGPT:

Based on the data provided, here are some possible requirements (scope) for developing a technology-oriented digital solution for the evolution of digitalization in the energy sector:

- Integration of renewable energy data: The digital solution should be able to integrate geographically dispersed energy data from different renewable energy sources.
- Integrated platform: The digital solution should provide an integrated platform for renewable energy companies to track assets, monitor energy production, and analyze data to meet the objective of decarbonization of the energy model.
- Real-time monitoring and predictive maintenance: The digital solution should provide real-time monitoring of renewable energy production and offer predictive maintenance alerts to prevent downtime.
- Accurate forecasting: The digital solution should offer accurate forecasting of weather and market conditions to maximize renewable energy production.
- Artificial intelligence and machine learning: The digital solution should use artificial intelligence and machine learning to optimize the engineering and construction of new renewable sources and plants.
- Digital twin technology: The digital solution should be backed by digital twin technology to help with modeling, forecasting, and testing for

optimal performance, from power generation to its link with the customers.

- Smart grid integration: The digital solution should integrate with existing power grids to create smart grids and improve energy efficiency.
- User connectivity: The digital solution should improve user connectivity and offer transparency in the management of renewable energy.
- Energy policy compliance: The digital solution should be designed to comply with energy policies and regulations.
- Competitive pricing: The digital solution should be cost-effective and offer competitive pricing for renewable energy.

2. List of tools, technologies and systems to support such needs:

Prompt provided in chatgpt :

Identify various technologies, tools and systems available in the market to support these needs/requirements.

ChatGPT:

Here are some technologies, tools, and systems available in the market that can support the needs/requirements for the evolution of digitalization in the energy sector:

- Internet of Things (IoT) devices: IoT devices such as sensors, smart meters, and controllers can be used to collect data from renewable energy sources and send it to a centralized platform for analysis.
- Big Data Analytics: Big data analytics tools can be used to analyze the vast amount of data generated by renewable

energy sources to improve energy efficiency and optimize power generation processes.

- **Machine Learning:** Machine learning algorithms can be used to predict energy demand and optimize energy usage for different applications.
- **Digital Twin Technology:** Digital twin technology can be used to create virtual replicas of renewable energy sources to model, forecast, and test optimal performance.
- **Blockchain:** Blockchain technology can be used to create a transparent and secure system for tracking renewable energy production and distribution.
- **Cloud Computing:** Cloud computing services can be used to store, process, and analyze large amounts of data generated by renewable energy sources.
- **Artificial Intelligence (AI):** AI can be used to optimize energy production and consumption by analyzing data from renewable energy sources and predicting demand.
- **Geographic Information System (GIS):** GIS tools can be used to visualize renewable energy sources, their production, and distribution, and to manage them more efficiently.
- **Smart Grid Technology:** Smart grid technology can be used to optimize energy consumption and reduce energy wastage by monitoring energy usage and providing feedback to users in real-time.
- **Energy Management Systems (EMS):** EMS can be used to monitor, control, and optimize energy usage in different applications, including homes, buildings, and industries.

3. Working API code:

Prompt provided in chatgpt :

Generate one API and suitable data analysis Code base to access

the energyrelated data set and perform data analysis.

ChatGPT:

I can provide you with a sample API and codebase for accessing and analyzing energy-related datasets. However, I would need to know more details about the specific dataset you would like to analyze and the purpose of the analysis in order to provide a fully customized solution. Nevertheless, here is a sample codebase and API for accessing and analyzing a hypothetical energy dataset

API:

```
import requests

def get_energy_data():
    # Make an API call to get the energy data
    url = "https://api.energydata.com/energydata"
    headers = {'Content-Type': 'application/json'}
    data = {"query": "energy", "limit": 1000}
    response = requests.get(url, headers=headers, json=data)

    # Check if the request was successful
    if response.status_code == 200:
        energy_data = response.json()
        return energy_data
    else:
        return None
```

In this code, we are making a GET request to the EnergyData API and passing the API key in the headers. We then return the response in JSON format.

Data Analysis Code:

```
import pandas as pd
import matplotlib.pyplot as plt

# Get the energy data using the API
energy_data = get_energy_data()

# Convert the JSON data into a Pandas DataFrame
energy_df = pd.DataFrame.from_records(energy_data['results'])

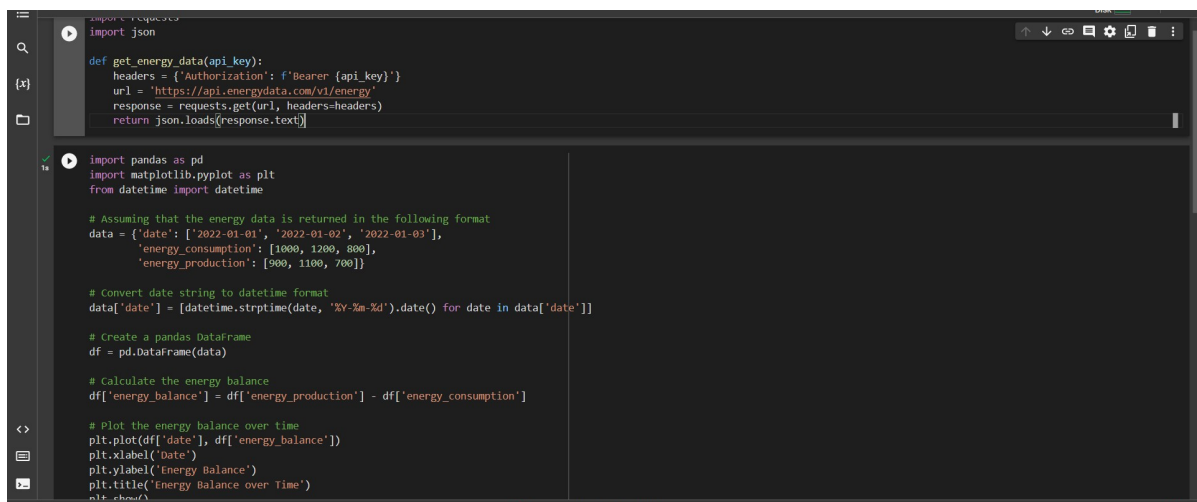
# Perform some basic data cleaning and processing
energy_df['date'] = pd.to_datetime(energy_df['date'])
energy_df.set_index('date', inplace=True)
```

```
# Plot a line graph of the energy consumption over time
plt.plot(energy_df['energy_consumption'])
plt.title('Energy Consumption over Time') plt.xlabel('Date')
plt.ylabel('Energy Consumption (kWh)') plt.show()

# Calculate the average energy consumption per day
daily_energy_consumption = energy_df['energy_consumption'].resample('D').sum()
avg_daily_energy_consumption = daily_energy_consumption.mean()
print(f"Average Daily Energy Consumption: {avg_daily_energy_consumption} kWh")
```

This code accesses the energy dataset through the API and converts the returned JSON data into a Pandas DataFrame. It then performs some basic data cleaning and processing and plots a line graph of the energy consumption over time. Finally, it calculates the average daily energy consumption and prints the result.

Executing the code



```
import requests
import json

def get_energy_data(api_key):
    headers = {'Authorization': f'Bearer {api_key}'}
    url = 'https://api.energydata.com/v1/energy'
    response = requests.get(url, headers=headers)
    return json.loads(response.text)

import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime

# Assuming that the energy data is returned in the following format
data = {'date': ['2022-01-01', '2022-01-02', '2022-01-03'],
        'energy_consumption': [1000, 1200, 800],
        'energy_production': [900, 1100, 700]}

# Convert date string to datetime format
data['date'] = [datetime.strptime(date, '%Y-%m-%d').date() for date in data['date']]

# Create a pandas DataFrame
df = pd.DataFrame(data)

# Calculate the energy balance
df['energy_balance'] = df['energy_production'] - df['energy_consumption']

# Plot the energy balance over time
plt.plot(df['date'], df['energy_balance'])
plt.xlabel('Date')
plt.ylabel('Energy Balance')
plt.title('Energy Balance over Time')
plt.show()
```

Data Analysis Plot:

