**CS301-Software Engineering**

Name: Chinmayee Patankar

Roll no.:21BCS079

Theme: Evolution of digitalisation in the energy sector:

The energy sector is now in a profound transition towards a very important energy transformation, and digitalisation is one of the key facilitators to ensure that it is fulfilled. In the recent past, companies started by switching the use of analogue meters to digital meters, smart meters etc., in order to improve energy efficiency.

Digitalisation acts as a lever in the sector to combat climate change and optimise power generation processes to reduce emissions and meet the objective of decarbonisation of the energy model.

Assignment:

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

a) Data Integration and Management: A digital solution that helps to collect, integrate and manage energy-related data from multiple sources such as renewable energy plants, smart grids, weather forecasts, and market conditions. This includes data from sensors, IoT devices, SCADA systems, and other sources.

b) Analytics and Predictive Maintenance: The solution must be able to analyse data in real-time and provide insights to improve the efficiency of power generation as well as reduce maintenance costs.

c) Security and Data Privacy: The solution must ensure the security and privacy of data collected and processed. It must follow the industry standards and regulations.

d) Scalability and Flexibility: The solution must be scalable to accommodate the growing needs of energy companies and flexible enough to adapt to changing energy policies and market conditions.

e) User Interface and Accessibility: The solution must have an easy user interface that is accessible to both technical and non-technical users. It should allow users to access data and insights from anywhere and anytime.

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

Open-source data integration tool like Apache Kafka, Apache NiFi, and cloud-based data integration services like Microsoft Azure Data Factory, Amazon Web Services (AWS) Glue, Google Cloud Data Fusion, and IBM Cloud Pak for Data can be used.

Hadoop, Apache Spark, TensorFlow, Keras, R, and Python can be used for data cleaning, exploratory data analysis, building and training machine learning models, and deploying models into production.

HashiCorp Vault, CyberArk, AWS Key Management Service (KMS), Google Cloud Key Management Service (KMS), and Microsoft Azure Key Vault are critical for securing sensitive data and ensuring that encryption keys and other secrets are properly managed and protected.

Docker, Kubernetes, Apache Mesos, and OpenStack are all popular open-source tools and platforms used in modern IT environments for managing and deploying applications and infrastructure.

Tableau, Microsoft Power BI, Google Data Studio, and Apache Superset are all popular open-source tools and platforms used in modern IT environments for managing and deploying applications and infrastructure.

1. Generate one API and suitable data analysis Code base to access the energy related data set and perform data analysis.

## We have two code files. The app.py file that has a Flask application where we have a route to render the graphs. Inside the route, we have the send\_from\_directory() function to send the image file to the client. Another file, the wind-solar-and-nuclear-energy-production.ipynb file has all code related to data analysis. Open the file in Jupyter notebook and run all cells. All the graphs will be saved in same directory. Move the image files to a directory accessible by the Flask application. You can create a “graphs” directory in your Flask app's static folder to store the images.

## Then run the flask application and open port localhost:5000 on your google chrome. Add /graphs/<filename> to the url. It will open the required graph.

## For ex. here I have used the query graphs/total\_solar\_production.png on url to get the below graph.

## 