**Why this tool?**

*To build asset config-data quality tool that highlights data quality gaps of Renewables Onshore - Wind asset configurations between RAMP, Predix and AWS asset model and drive necessary actions in the data curation stage.* The asset and customer details are maintained on different data sources like RAMP, GR, Predix application; where RAMP is a repository which has more than 55,000+ assets and it acts as the source of truth for other asset models. The data maintained on RAMP flows into the Predix application via an edge server.

Since the world is migrating into AWS, there is need to build the AWS asset model, and the data from RAMP must be ingested into AWS. The data between Predix, RAMP and AWS needs to be in sync. But these ingestions have caused a few data quality gaps like data discrepancies and missing data from either of the data sources.

We are coming up with a tool that highlights the data quality gaps between RAMP, Predix and AWS asset model and based on the tool’s output the end users will drive necessary actions for the data curation. Discrepancy tracking in this tool is done by identifying the key asset config parameters like hub height, altitude, frequency, air density, rating, site name, pad number, model, make, cod, rotor diameter to compare the asset details and generate a discrepancy list.

**What the Tool Does?**

* The end user of the tool would have to upload an excel sheet with the attribute values AWS, RAMP and Predix. The sheet should consist of the turbine ID and its corresponding attribute value from AWS, RAMP and Predix respectively.
* The user should then choose the corresponding attribute from the given options in the tool. Each attribute selection would trigger its own internal logic to finally generate the discrepancy list.
* On submitting, the corresponding attribute’s internal logic would access the data values of the asset model and compare them.
* Once the comparison of the attributes values from the data sources is done, a new sheet in the same workbook would be generated with a name "output\_date". This sheet would have the ID and attribute value from AWS, RAMP and Predix and its discrepancy status, i.e., if the values are matching or not matching or if the ID is missing in either of the data sources.
* Once this is done the user is given an option to carry out the same action for other attributes or can also exit if the user wishes to.

**Technologies used to develop this tool:**

* The front-end is developed using **HTML**, **CSS**, and **Bootstrap**.
* The back end is developed using **Flask**, which is a web framework written in python.
* Python Libraries like **openpyxl, matplotlib, NumPy, pynput, pandas, colorama** are used to access the data in excel and insert formulas like **VLOOKUP, TRIM, IF-ELSE, ISNA, DATE, TEXT, FILTER,ROUND** to compare the data and generate the discrepancy list. The libraries of flask like **Flask, render\_template, request, redirect, url\_for** are used for the backend operations.
* The platform used to set up this tool is VSCode.

**Logic for the comparison:**

* **Non-Repowered Attributes**:
  + The input Excel file would have ID and the attribute value in the order AWS, RAMP followed by SourceKey, Serial Number and Attribute value from Predix.
  + The Tool fetches the attribute value from RAMP and Predix with respect to the AWS ID.
  + Then the values from RAMP and Predix are compared against AWS.
  + If either of the IDs are absent in any of the data sources they are marked as “**RAMP id not in AWS**” or “**AWS id not in RAMP**” or “**Predix id not in AWS**” or “**AWS id not in Predix**”.
  + While Comparing AWS against Predix, the presence of AWS ID is checked in SourceKey list, if it is not present then we check for the presence of AWS ID in SerialNumber list. If the ID is present in either in SourceKey or SerialNumber list then the tool fetches the attribute value from Predix with respect to AWS ID else, it is marked as AWS id not in Predix.
  + Finally, these comparisons are put into the new sheet “**output\_date**” in a consolidated format.
* **Repowered Attributes**:
  + The input Excel file would have ID and the attribute value in the order AWS, RAMP followed by SourceKey, Serial Number and Attribute value from Predix.
  + The Tool fetches the attribute value from RAMP and Predix with respect to the AWS ID.
  + Then the values from RAMP and Predix are compared against AWS.
  + If either of the IDs are absent in any of the data sources they are marked as “**RAMP id not in AWS**” or “**AWS id not in RAMP**” or “**Predix id not in AWS**” or “**AWS id not in Predix**”.
  + In the input sheet the normal attributes should be placed before the repowered attributes(**ie. attribute, R\_attribute**).
  + While comparing the repowered attributes, if the discrepancy status of **R\_attribute**s is “**matching**” then the normal attributes are not compared. In all other cases the normal attributes are also compared.
  + While Comparing AWS against Predix, the presence of AWS ID is checked in SourceKey list, if it is not present then we check for the presence of AWS ID in SerialNumber list. If the ID is present in either in SourceKey or SerialNumber then the tool fetches the attribute value from Predix with respect to AWS ID else, it is marked as AWS id not in Predix.
  + Finally, these comparisons are put into a new sheet in a consolidated format. The tool first places the repowered attribute value comparisons and then the normal attribute value comparisons.
* **Multiple Attributes**:
  + The input Excel file would have separate sheets “AWS”, “RAMP” and “Predix” with ID and the attribute values from the corresponding data sources.
  + The attributes need to be in a specific order for the multiple attribute comparison.
    - **AWS**: \_id , Grid\_frequency , Hub\_height , Reference\_airdensity , Elevation , Make , Rotor\_diameter , Short\_name , Rated\_power , Commercial\_operation\_date , Model , Properties\_name.
    - **RAMP**: Turbine\_serial\_number , Frequency , Tower\_height , Air\_density , Altitude , OEM\_supplier , Rotor\_diameter , Pad\_number , R\_new\_rating , Rating , R\_COD\_date , COD\_date , Current\_model\_name , Model\_name , Site\_name.
    - **Predix**: SourceKey , SerialNumber , Frequency , Hub Height , Reference Air Density , Location.altitude , Rotor Diameter , Pad Number , Rated Output , Model , Name.
  + The Tool fetches the attribute value from AWS and RAMP with respect to the AWS ID into a new sheet **op\_date\_(AWS vs RAMP)**. It also fetches the attribute values from AWS and Predix with respect to AWS ID into a new sheet **op\_date\_(AWS vs Predix)**.
  + Then the values from RAMP and Predix are compared against AWS in the corresponding sheets with color coding.
  + A separate column in this sheet highlights the overall discrepancy of the row.
  + If either of the IDs are absent in any of the data sources they are marked as “**RAMP id not in AWS**” or “**AWS id not in RAMP**” or “**Predix id not in AWS**” or “**AWS id not in Predix**”.
  + While Comparing AWS against Predix, the presence of AWS ID is checked in SourceKey list, if it is not present then we check for the presence of AWS ID in SerialNumber list. If the ID is present in either in SourceKey or SerialNumber then the tool fetches the attribute value from Predix with respect to AWS ID else, it is marked as AWS id not in Predix.