Project name: Data Analyst:: Data Collection Pipeline (Data Acquistion to Storytelling) - Data Collection Pipeline (Data Acquisition to Story telling)

Debika Mukherjee (Solo project)
Email: debika.isms@gmail.com
Country: United Kingdom
University: Cardiff metropolitan university
Specialization: Data Analyst

Problem description:
XYZ company is collecting the data customer using google forms/survey monkey and they have floated n number of forms on the web.
Company wants to create a pipeline which will collect all the data of these google forms/survey monkey and visualize the data in the dashboard.
Company wants clean data and if there is any data issue present in the data then it should be treated by this pipeline (duplicate data or junk data). dedup check should be performed on the email id of the customer

Github repository:
debikaisms/Final-project: Data Collection Pipeline (Data Acquistion to Storytelling) Data Collection Pipeline (Data Acquisition to Story telling) (github.com)

Data cleansing and transformation done on the data.

```
pip install pandas openpyxl


import pandas as pd

# Specify the path to your Excel file
excel_file_path = 'surveydata.xlsx'

# Read data from Excel into a DataFrame
#df = pd.read_excel(\content\surveydata.xlsx, sheet_name='Sheet1')
df = pd.read_excel(excel_file_path, sheet_name='Sheet3')


# 1. Duplicate Removal based on Email Address
df = df.drop_duplicates(subset='Email Address', keep='first')

# 2. Data Type Conversion (if necessary)
# Example: df['Age'] = df['Age'].astype(int)

# 3. Handling Missing Values (if necessary)
# Example: df.dropna(subset=['column_name'], inplace=True)

# 4. Text Cleaning (if necessary)
# Example: df['Product Feedback'] = df['Product Feedback'].str.lower()

# 5. Outlier Detection and Handling (if necessary)
# Example: Identify and handle outliers in numeric columns

# 6. Standardization (if necessary)
# Example: Standardize values in the 'Gender' column

# 7. Aggregation (if necessary)
# Example: Calculate aggregate statistics
```

```
# Save the cleansed and transformed data back to Excel
df.to_excel('cleansed_data.xlsx', index=False, engine='openpyxl')


df['Age'].fillna(df['Age'].mean(), inplace=True)
```