Shega

# Data Engineering Intern Assignment

### Case Study: Weather Data Processing Pipeline

**Objective:** Build a simple data pipeline to ingest, clean, transform, and analyze weather data from a CSV file, saving the processed results as output files.

**Tools and Technologies:**

- **Programming Language:** Python (using Pandas and NumPy)
- **Output Format:** CSV or JSON files

**Requirements:**

1. **Data Ingestion:**
    a. Start with a CSV filenamed “wather_data.csv” containing weather data with columns like `date`, `city`, `temperature_celsius`, `humidity_percent`, `wind_speed_kph`, and `weather_condition`. This file is should be attached to this document.
    b. Load this CSV into a Pandas DataFrame using Python.
2. **Data Cleaning and Transformation:**
    a. **Handle Missing Values:** For example, replace missing `temperature_celsius` values with the average temperature for that city, or drop rows where `date` is missing.
    b. **Standardize Dates:** Convert the `date` column to a consistent format (e.g., YYYY-MM-DD).
    c. **Add a New Column:** Create a `temperature_fahrenheit` column by converting `temperature_celsius` using the formula: $F=C\times 9/5+32$
    d. **Filter Data:** Keep only rows where `weather_condition` is not "Unknown" or null.
3. **Data Output:**
    a. Save the cleaned and transformed data (including the new `temperature_fahrenheit` column) as a CSV file named “tranformed_weather_data.csv” under “outputs” folder.
    b. **Optional:** Generate a simple text report (e.g., Markdown or TXT file) listing the top 5 cities with the highest average `temperature_celsius`.
4. **Deliverables:**
    a. A GitHub repository containing:

i. Python script(s) for data ingestion, cleaning, transformation, and output generation.
ii. Output files (CSV/JSON and optional report).
iii. A README file including:
    1. Step-by-step instructions to run the pipeline locally.
    2. A brief explanation of your approach and any challenges faced.
    3. (Optional) Sample output or visualization if you complete the bonus task.

b. **Bonus (Optional):** Create a bar chart of average temperature per city using Matplotlib or Seaborn and include the image in the repository.

**Evaluation Criteria:**

- **Code Quality:** Clean, well-structured, and readable Python code with effective use of Pandas.
- **Data Handling:** Correct ingestion, cleaning, and transformation of the weather data.
- **Output Files:** Properly formatted CSV/JSON files meeting the requirements.
- **Problem-Solving:** Smart handling of data issues and adherence to the specified tasks.
- **Documentation:** Clear, concise instructions and explanations in the README.

**Time Estimate:** 4-6 hours

**Difficulty:** Easy – Focuses on fundamental data engineering skills without database complexity.