How to run this ETL:

1) run the Snowflake query to create the **stages** for S3 buckets with your S3 credentials
2) Put your credentials for S3 and snowflake in the python script named Dublin_Transportation_ETL.py (you can get this information in the amazon web portal related cloud services and the snowflake web portal)

   S3:

```
aws_access_key = "xxxxxx"
aws_secret_key = "xxxxxxxx"
region = "us-east-2"
file_url = CSV_URL["Weather_Data_Met_Eireann"]
excel_csv_data = load_excel_from_url(EXCEL_URL["Cycle_Counts"])
bucket_name = "s3carloslinaresk81"
s3_file_name = f"Weather_Data_Met_Eireann_{formatted_datetime}.csv"
s3_excel_file_name = f"Cycle_Counts_{formatted_datetime}.csv"
folder_name = "s3_bucket_folder"
s3_excel_folder_name = "s3_excel_bucket_folder"
s3_folder_file_name = f"{folder_name}/{s3_file_name}"
s3_excel_folder_file_name = f"{s3_excel_folder_name}/{s3_excel_file_name}"
```

   Snowflake:

```
SNOWFLAKE_CONFIG = {
    "user": "xxxxxx",
    "password": "xxxxx",
    "account": "xxxx",
    "warehouse": "COMPUTE_WH",
    "database": "ETL_DATABASE",
    "schema": "ETL_SCHEMA",
    "role": "ACCOUNTADMIN"
}
```

3) Run the python script named Dublin_Transportation_ETL.py

   Notes:

   A) Access keys and secret key were used for this exercise for ease of use, but there are more secure methods that should be implemented on a prod, QA and develop environment

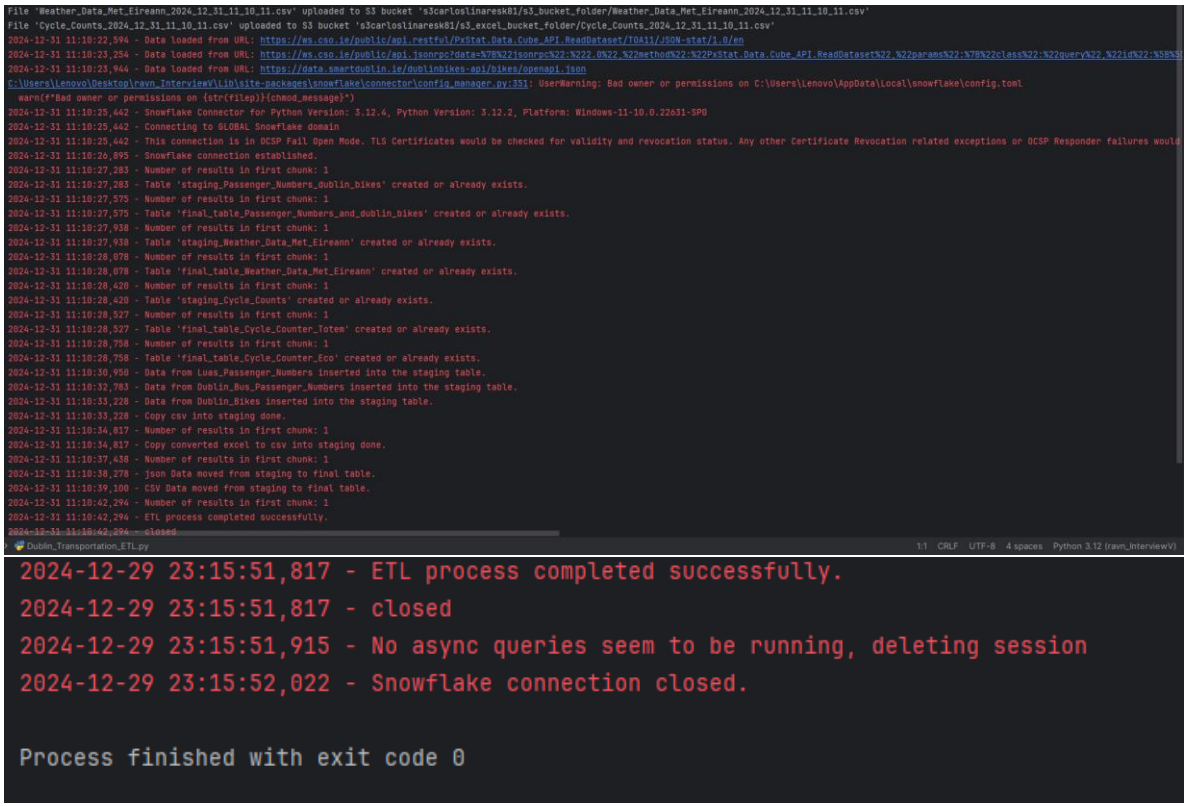
   B) The daily run job in configure in Cron using this script:

   Crontab -e to open a crontab in Linux

```
GNU nano 6.2                                                    /tmp/crontab.noq2f0/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 7 * * * /mnt/c/Users/Lenovo/Desktop/Porfolio/Scripts/python.exe /mnt/c/Users/Lenovo/AppData/Roaming/Jetbrains/pycharmCE2024.1/scratches/Dublin_Transportation_ETL.py
```

On the lines that are not commented are the path of your python.exe is specified followed by a space and then the path of the script to be executed.

C) A successful run of the python script looks like this:



D) Json files were loaded from the URL without using S3 because it's not a heavy load. On the other hand, the .CSV are a much heavier load so that's why S3 was used with the copy into command that's optimized for these scenarios. Finally, the excel file needed to be converted to .csv because snowflake works better with csv instead of .xlsx format files.