import glob

import pandas as pd

import xml.etree.ElementTree as ET

from datetime import datetime

log\_file = "log\_file.txt"

target\_file = "transformed\_data.csv"

def extract\_from\_csv(file\_to\_process):

    dataframe = pd.read\_csv(file\_to\_process)

    return dataframe

def extract\_from\_json(file\_to\_process):

    dataframe = pd.read\_json(file\_to\_process, lines=True)

    return dataframe

def extract\_from\_xml(file\_to\_process):

    dataframe = pd.DataFrame(columns=["name", "height", "weight"])

    tree = ET.parse(file\_to\_process)

    root = tree.getroot()

    for person in root:

        name = person.find("name").text

        height = float(person.find("height").text)

        weight = float(person.find("weight").text)

        dataframe = pd.concat([dataframe, pd.DataFrame([{"name":name, "height":height, "weight":weight}])], ignore\_index=True)

    return dataframe

def extract():

    extracted\_data = pd.DataFrame(columns=['name','height','weight']) # create an empty data frame to hold extracted data

    # process all csv files

    for csvfile in glob.glob("\*.csv"):

        extracted\_data = pd.concat([extracted\_data, pd.DataFrame(extract\_from\_csv(csvfile))], ignore\_index=True)

    # process all json files

    for jsonfile in glob.glob("\*.json"):

        extracted\_data = pd.concat([extracted\_data, pd.DataFrame(extract\_from\_json(jsonfile))], ignore\_index=True)

    # process all xml files

    for xmlfile in glob.glob("\*.xml"):

        extracted\_data = pd.concat([extracted\_data, pd.DataFrame(extract\_from\_xml(xmlfile))], ignore\_index=True)

    return extracted\_data

def transform(data):

    '''Convert inches to meters and round off to two decimals

    1 inch is 0.0254 meters '''

    data['height'] = round(data.height \* 0.0254,2)

    '''Convert pounds to kilograms and round off to two decimals

    1 pound is 0.45359237 kilograms '''

    data['weight'] = round(data.weight \* 0.45359237,2)

    return data

def load\_data(target\_file, transformed\_data):

    transformed\_data.to\_csv(target\_file)

def log\_progress(message):

    timestamp\_format = '%Y-%h-%d-%H:%M:%S' # Year-Monthname-Day-Hour-Minute-Second

    now = datetime.now() # get current timestamp

    timestamp = now.strftime(timestamp\_format)

    with open(log\_file,"a") as f:

        f.write(timestamp + ',' + message + '\n')

# Log the initialization of the ETL process

log\_progress("ETL Job Started")

# Log the beginning of the Extraction process

log\_progress("Extract phase Started")

extracted\_data = extract()

# Log the completion of the Extraction process

log\_progress("Extract phase Ended")

# Log the beginning of the Transformation process

log\_progress("Transform phase Started")

transformed\_data = transform(extracted\_data)

print("Transformed Data")

print(transformed\_data)

# Log the completion of the Transformation process

log\_progress("Transform phase Ended")

# Log the beginning of the Loading process

log\_progress("Load phase Started")

load\_data(target\_file,transformed\_data)

# Log the completion of the Loading process

log\_progress("Load phase Ended")

# Log the completion of the ETL process

log\_progress("ETL Job Ended")