DA108 | Lab 07 Assignment

Objective A: File Management System for a Research Lab

You are working as a data manager in a research lab that collects and organizes data from various sources. Your task is to automate the organization of files stored in a messy directory containing text files (.txt), CSV datasets (.csv), JSON records (.json), images (.jpg, .png), and log files (.log).

Step 1 - Understanding the Data Directory

- Given a dataset directory (data_repository/), first, list all files using Python.
- Print the total number of files in the directory.

Step 2 - Categorizing Files

- Use os and glob to identify and count files by type (TXT, CSV, JSON, JPG, PNG, LOG).
- Print the count of each file type.

Step 3 - Organizing Files into Folders

- Create subdirectories inside organized_data/ for each file type (e.g., text_files/, csv_files/, images/, logs/).
- Move files into their respective folders.

Step 4 - Handling Missing Directories

- If a required subdirectory does not exist, create it dynamically.
- Ensure no files are lost in the process.

Step 5 - Renaming and Timestamping Files

- Append a timestamp (YYYYMMDD_HHMMSS) to log files before moving them (e.g., server_20250221_153045.log).

Step 6 - Generating a Summary Report

- Write a Python script to generate a summary (summary.txt) containing:
    - Number of files per category.
    - Total number of files moved.
    - A sample file name from each category.

Objective B: Experimenting with loading files

You have joined a data analytics team that works with a variety of file formats. Your first task is to read, extract basic information, and print summaries from different types of files stored in a directory dataset.

Step 1 - Understanding the Data Directory

Imagine the provided dataset/ folder contains files of various formats:

- Text file (data.txt)
- CSV file (data.csv)
- JSON file (data.json)
- Excel file (data.xlsx)

- Image file (image.jpg)
- PDF file (document.pdf)

Write a script to list all files in the directory and print their names.

Step 2 - Opening and Reading Files

Implement functions to open and read each file type:

Text File (.txt) → Read and print the first 5 lines.
CSV File (.csv) → Use csv or pandas package to load and display the first 3 rows.
JSON File (.json) → Use json package to parse and display key-value pairs.
Excel File (.xlsx) → Use openpyxl or pandas to read and display the first 3 rows.
Image File (.jpg) → Use PIL to open and display the image.
PDF File (.pdf) → Use PyPDF2 or pdfplumber to extract and print the first few lines.

Step 3 - Handling Errors Gracefully
Implement exception handling to manage cases where:

- A file might be missing.
- The required package is not installed.
- The file is corrupted or unreadable.