| **Project Title:** | **A Comprehensive ETL Workflow with Python for Data Engineers** |
| --- | --- |
| **Technologies** | **Python** |
| **Data Format** | **CSV , JSON , XML** |

### 

### 

### **Project Orientation video :** [**Project Orientation - comprehensive ETL**](https://drive.google.com/file/d/1Ln6aYZmABzDa68PXGygRT_oJSoaQd9qj/view?usp=sharing)

### **Introduction:**

The Extract, Transform, Load (ETL) process is essential for data engineers, enabling them to manage data from various formats and transform it for further use. In this project, we will demonstrate how to extract data from CSV, JSON, and XML formats, transform it, and load the transformed data into a structured format for further processing.

### **Objectives:**

By the end of this project, you will be able to:

1. Extract data from CSV, JSON, and XML files.
2. Transform the extracted data into a desired format, including unit conversions.
3. Load the transformed data into a CSV file for future use in databases.
4. Log the progress of ETL operations for monitoring purposes.

### **Dataset :**wget <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/source.zip>

### **Steps:**

#### **Step 1: Gather Data Files**

1. Open a terminal and download the dataset:
   * Use the wget command to download the dataset containing multiple file formats.

### wget <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/source.zip>

1. Unzip the downloaded file:

Expand-Archive -Path source.zip -DestinationPath ./unzipped\_folder

Note : You may use the unzip command to extract the contents of the downloaded zip file.

After completing this step, the project folder will have CSV, JSON, and XML files to work with.

#### **Step 2: Import Libraries and Set Paths**

You need to import necessary libraries like:

* **glob** to handle file formats.
* **pandas** to read CSV and JSON files.
* **xml.etree.ElementTree** to parse XML data.
* **datetime** to track the progress of each phase through logging.

Install the **pandas** library if it's not already installed.

Set up paths for:

* **log\_file.txt** to record the logs.
* **transformed\_data.csv** to save the final output.

#### **Step 3:Define functions for each step of ETL as follows:**

1. **Extract Data:**
   * Three different functions to extract data from CSV, JSON, and XML files respectively.
   * A master function will call the relevant function based on the file type and combine the extracted data into a single DataFrame.
2. **Transform Data:**
   * The transformation process involves converting:
     + Heights from inches to meters.
     + Weights from pounds to kilograms.
   * This step ensures the data is in the desired format for further analysis or storage.
3. **Load Data:**
   * The transformed data is saved to a CSV file, which can later be loaded into a relational database or used for further processing.
4. **Logging:**
   * Throughout the ETL process, each phase (Extraction, Transformation, Loading) is logged with a timestamp to ensure traceability and monitoring.
   * Logs are saved in a text file for auditing or troubleshooting purposes.

### **Step 4: ETL Execution**

The ETL process follows this sequence:

1. **Extraction Phase:**
   * The project extracts data from all CSV, JSON, and XML files located in the project directory.
   * Each file type is processed, and the results are combined into one DataFrame.
2. **Transformation Phase:**
   * The extracted data undergoes transformation to convert the measurements to standard units (e.g., height to meters, weight to kilograms).
3. **Loading Phase:**
   * The transformed data is written into a CSV file, which can be imported into a database for further use.
4. **Logging:**
   * The start and end of each phase (Extraction, Transformation, Loading) are logged to track progress and ensure everything runs smoothly.

### **Conclusion:**

This project highlights the practical implementation of ETL processes using Python. The data extraction from multiple file formats, transformation of units, and loading of the final data into a structured CSV format demonstrate essential data engineering skills. Additionally, by logging each step of the process, you can monitor the progress and debug issues if they arise

**Submission:**

* Provide a well-commented Python file containing the complete code for the project, organized into sections for data Pipeline and Analysis.
* Upload the same into github with a proper Readme file.
* Presentation on the entire project, including Problem Statement, Tools Used, Approaches and Insights Found.

**Evaluation Metrics:**

* Project evaluation will be done in the live session and have to showcase the approaches done to complete the project
* Secure code : follow the secure code guidelines
  + [Special session for Secure Code: Managing Credentials with Your Python Script(28/09/2024)](https://docs.google.com/document/d/1lJcPv5VGktqTl7SygS9Ot14zHCy7OjrXjoFSIDTFeMM/edit?usp=sharing)
* You are supposed to write a code in a modular fashion (in functional blocks)
* Maintainable: It can be maintained, even as your codebase grows.
* Portable: It works the same in every environment (operating system)
* You have to maintain your code on GitHub.(Mandatory)
* You have to keep your GitHub repo public so that anyone can check yourcode.(Mandatory)
* Proper readme file you have to maintain for any project development(Mandatory)
* Follow the coding standards:
  + https://www.python.org/dev/peps/pep-0008/
* You should include basic workflow and execution of the entire project in the readme file on GitHub

**GitHub Repo:**

The attached reference document will help you use GitHub effectively. - [Link](https://docs.google.com/presentation/d/1XHCbgUOqbcXNUyQ87vTlKdKRgAbBxtkA/edit?usp=sharing&ouid=106590842700357786537&rtpof=true&sd=true)

**Reference Material:**

Official Documentation:

* <https://www.python.org/doc/>

PROJECT DOUBT CLARIFICATION SESSION ( PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.

Note: Book the slot at least before 12:00 Pm on the same day

Booking link :<https://forms.gle/NtkQ4UV9cBV7Ac3C8>

Session timing: 04:00 pm to 5:00 pm every saturday

================================================================

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

Note: This form will Open on Saturday (after 2 PM ) and Sunday Only on Every Week

Session Timing: Monday-Saturday (11:30AM to 1:00PM)

Booking link :<https://forms.gle/1m2Gsro41fLtZurRA>