**ETL Documentation**

**ETL Pipeline Project Overview**

**Objective:** Developed a Python-based ETL pipeline to centralise personal data from multiple sources into a MySQL database, aiming to enhance data accessibility for future application development.

**Technologies Used:** Python, MySQL, API integration, HTML scraping.

**Key Features:** Implemented diverse data extraction techniques (API calls, manual exports, HTML scraping), developed a modular pipeline for scalable processing, and integrated data from various data formats (Excel, CSV, JSON, XML)into 19 distinct MySQL tables.

**Outcome:** Established a comprehensive personal data repository, laying the foundation for developing personalized applications and dashboards with the ability to query historical data.

**Project Summary**

**Directory Structure**

├── \_\_init\_\_.py # Marks directory as Python package directory.

├── .env # Environment variables file.

├── .git # Git repository metadata and tracking information.

├── .gitignore # Lists files and directories ignored by Git.

├── archive # Folder for archived items.

├── config.py # Configuration settings for the project.

├── constants.py # Constants used across the project.

├── credentials # Stores sensitive credentials (should be ignored by Git).

├── data # Data files for the project (consider ignoring large or sensitive data).

├── documentation # Project documentation.

├── exploratory\_analysis # Scripts/notebooks for exploratory data analysis.

├── extractors # Modules/scripts for data extraction.

├── loaders # Modules for loading data into the system or database.

├── main.py # Main script for running the project.

├── requirements.txt # Project dependencies.

├── setup\_cron.sh # Shell script for setting up cron jobs.

├── tests # Test scripts and test data.

├── transformers # Modules for data transformation.

├── utility # Utility scripts and helper functions.

├── validation # Validation scripts or modules for data or inputs.

**Data**

* **Apple Health**: Includes datasets like walking metrics, daily activity, blood glucose levels, heart rate, fitness metrics, low heart rate events, running metrics, sleep analysis, and steps.
* **Strava**: Sports activity data covering performance metrics, sport types, gear information, and activity details.
* **YouTube**: Data related to YouTube likes/dislikes, subscriptions, etc.
* **Daylio**: Mood and activity tracking information.
* **Spend**: Financial data personally tracked over 6 years

**Key Components**

- **Extractors**: Modules to retrieve data from various sources.

- **Transformers**: Modules to clean and standardise the data format.

- **Loader**: Modules responsible for inserting the data into the MySQL database.

- **Utility**: Contains helper functions and classes for database interactions (`DatabaseHandler`) and file management (`FileManager`).

- **Validation**: Script for post-load data validation to ensure data integrity and consistency.

**Usage**

**Installation**

**1. Clone the Repository**

git clone https://github.com/yourusername/yourproject.git

cd yourproject

**2. Install Dependencies**

Ensure Python 3.9+ is installed, then run:

pip install -r requirements.txt

**Manual Execution**

To execute the ETL process manually, run the following command in your terminal:

python main.py

**Automation**

For automated execution, set up scheduling tasks on your operating system. This can be achieved through cron jobs on Unix-based systems or Task Scheduler on Windows.