**1. Name of program:**

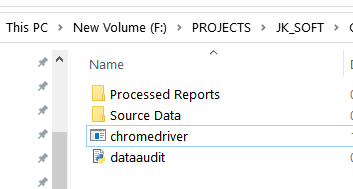
Operational Process Automation – Reporting.

**2. Objective:**

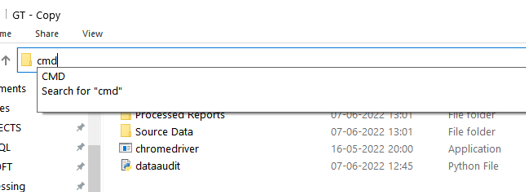
A code-based solution using Python and SQL that will automate the process of importing data, preparing reports, and delivering reports on/at scheduled days/times, to designated stakeholders.

**3. Scope – Summary:**

* + Project - Requirements
  + Folder Structure Creation for the program flow
  + Source file creation/updation with current date
  + Running Python File
  + Program-Execution:
    - Read the source files for data
    - Export the read data to DBMS as per the provided mappings.
    - Report generation from DBMS to a new excel file with necessary customizations.
  + **Project – Requirements**
  + Python – Version 3.9.13
  + Chrome Driver – To automate browser to get Visa bulletin Information.
  + Pyodbc – To establish connection between Python code and DBMS
  + Pandas – To retrieve and manipulate source file data
  + Chardet – for encoding
  + Xlsxwriter – to write data from data frames to external excel files
  + Openpyxl – for excel formatting
  + Email – to enabling E-mailing functionality from script
  + Fernet – for data encryption
  + **Folder Structure for the program flow:**
  + The following folders will be manually created at the path where the code will be placed for the ease of flow of program
  + Processed Reports
  + Source Data

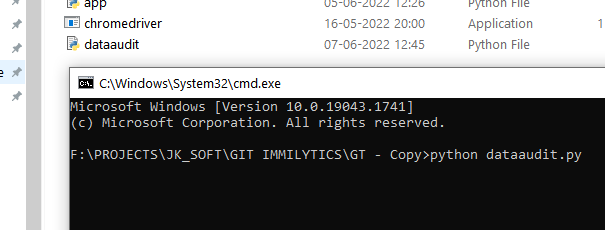


* Processed Reports folder - The path that will hold the processed output files
* Source data folder - The path that will hold the Source input files which are going to be processed.
* **dataaudit.py** - This is the executable Python file that we have to run it in the command prompt.
* **Running a Python File:**
* Go to the respective folder and type **cmd** in the address bar and click enter it will open a command prompt or else you can open command prompt from the run command (Win Key + R) and move to the respective folder by using **cmd command.**

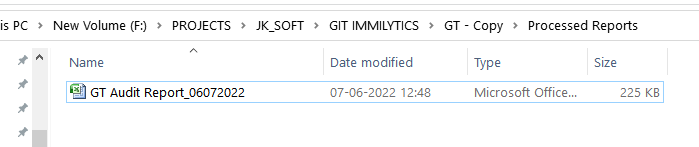
****

* Then run python **dataaudit** file by running following command:

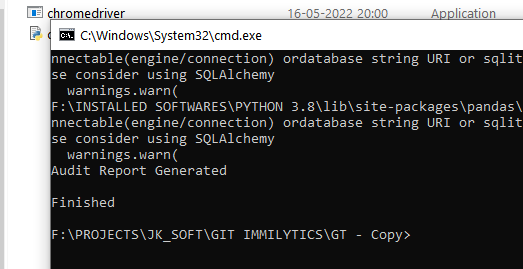
python **dataaudit**.**py**



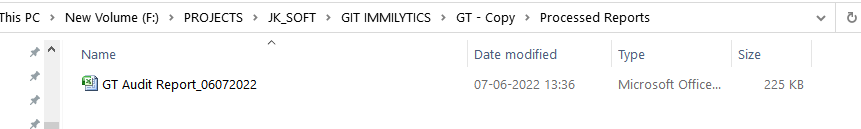
* And Then Press Enter to start Program Execution.
* **Program-Execution:**
* **Reads the source files for data (Beneficiary File):**
* Thus the acquired source files namely Reports Automation - Beneficiary Data (HQ Based) (for ImmiLytics), Reports Automation - Case Data (HQ Based) will be renamed as per the below conventions
* Reports Automation\_Beneficiary Data\_(client-name)\_mmddyyyy
* Reports Automation\_Case Data\_(client-name)\_mmddyyy and will be placed in the source file folder.
* Now the code searches and picks for the Beneficiary data file in the source folder that contains the current date in its filename.
* This uses pandas framework to read the content of the file and stores within as data frames
* Now the code reads the number of rows the dataframe has and loops repetitive for ‘n’ number of rows the dataframe holds.
* During each looping it reads the data of all the columns of the each respective rows and assigns to the respective columns as defined by the mappings and pushes/adds the data of each rows to the Beneficiary table as denoted in the respective SQL code.
* The loop ends once feeding data of all the rows in dataframe that is extracted from the source beneficiary file.
* **Reads the source files for data (Case data File):**
* Once the insertion of data to the DBMS from the beneficiary data is completed the same identical process is restarted for case file as elaborated below
* The code now searches and picks for the Case data file in the source folder that contains the current date in its filename.
* This uses pandas framework to read the content of the file and stores within as data frames.
* Now the code reads the number of rows the dataframe has and loops repetitive for ‘n’ number of rows the the dataframe holds.
* During each looping it reads the data of all the columns of the each respective rows and assigns to the respective columns as defined by the mappings and pushes/adds the data of each rows to Case table as denoted in the respective SQL code with the respective beneficiaryXref as ForeignKey.
* The loop ends once feeding data of all the rows in dataframe that is extracted from the source case file.
* **Report generation:**
* Now once all the datas from the source files are available in the DBMS it’s time to query all the tables as per the requirements to create a output file as required.
* The queried data are formatted and fed to create a Output file as below with the following tabs in it:
* Open Cases -
* Approved & Closed Cases
* Active Beneficiary List



* **Sending mail:**
* Once all the reports are ready it’s time to mail to the Respective e-mail.
* The code checks for availability of the recipient mail id that is provided within the code and further checks if the SEND\_MAIL option is enabled/disabled
* The code gets the mail ID’s of the primary recipient and secondary recipient from the code and goes forward to mail only when SEND\_MAIL option is enabled
* **Processed Reports:**
* Once processed, the command prompt will show ‘Finished’ status. Upon which we shall check the processed reports.



* The Processed Report will be created at Processed Reports Folder with the respective date of report generation.



-------------------------------------------------------------------------------