PER SCHOLAS

CAP 350 - Data Engineering - Capstone Project Requirements Document

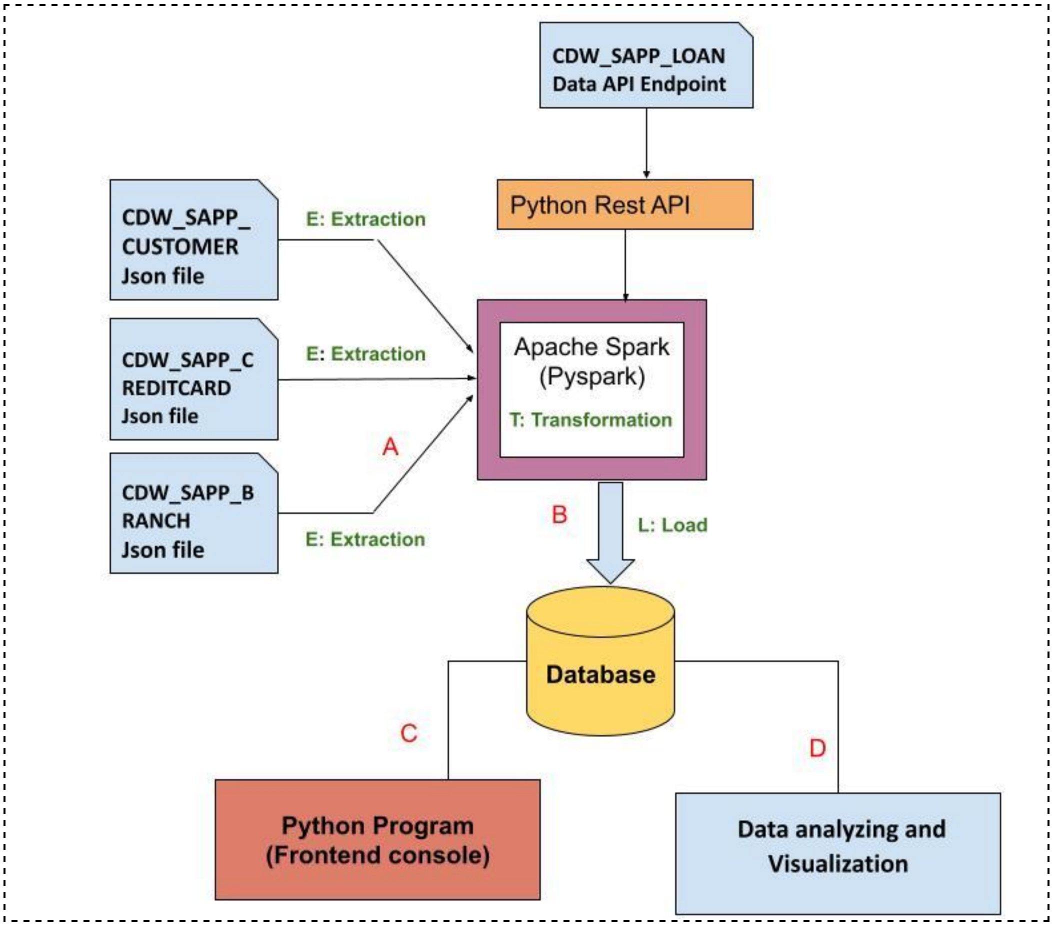
Overview:

This capstone project is your opportunity to demonstrate the knowledge and abilities you have acquired throughout the course.

This Capstone Project requires learners to work with the following technologies to manage an ETL process for a **Loan Application dataset** and a **Credit Card dataset**: Python (Pandas, advanced modules, e.g., Matplotlib), SQL, Apache Spark (Spark Core, Spark SQL), and Python Visualization and Analytics libraries. Learners are expected to set up their environments and perform installations on their local machines.

Workflow Diagram of the Requirements.

The workflow below will help you understand the application's requirements and flow.



Credit Card Dataset Overview.

The Credit Card System database is an independent system developed for managing activities such as registering new customers and approving or canceling requests, etc., using the architecture.

A credit card is issued to users to enact the payment system. It allows the cardholder to access financial services in exchange for the holder's promise to pay for them later. Below are three files that contain the customer's transaction information and inventories in the credit card information.

a) **CDW_SAPP_CUSTOMER.JSON:** This file has the existing customer details.
b) **CDW_SAPP_CREDITCARD.JSON**: This file contains all credit card transaction information.
c) **CDW_SAPP_BRANCH.JSON:** Each branch's information and details are recorded in this file.

Click here to download the Credit Card system files.

Business Requirements - ETL

1. Functional Requirements - Load Credit Card Database (SQL)

Req-1.1	**Data Extraction and Transformation with Python and PySpark**
Functional Requirement 1.1	a) For **“Credit Card System,”** create a Python and PySpark SQL program to **read/extract** the following JSON files **according to the specifications found in the** mapping document**.** 1. CDW_SAPP_BRANCH.JSON 2. CDW_SAPP_CREDITCARD.JSON 3. CDW_SAPP_CUSTOMER.JSON **Note: Data Engineers will be required to transform the data based on the requirements found in the** Mapping Document. Hint: [You can use PYSQL “select statement query” or simple Pyspark RDD].
Req-1.2	**Data loading into Database**
Function Requirement 1.2	Once PySpark reads data from JSON files, and then utilizes Python, PySpark, and Python modules to load data into RDBMS(SQL), perform the following: a) Create a Database in SQL(MySQL), named **“creditcard_capstone.”** b) Create a Python and Pyspark Program to load/write the “Credit Card System Data” into RDBMS(**creditcard_capstone**). Tables should be created by the following names in RDBMS: ***CDW_SAPP_BRANCH*** ***CDW_SAPP_CREDIT_CARD*** ***CDW_SAPP_CUSTOMER***

2. Functional Requirements - Application Front-End

Once data is loaded into the database, we need a front-end (console) to see/display data. For that, create a **console-based Python program** to satisfy System Requirements 2 (2.1 and 2.2).

2.1 Transaction Details Module

Req-2.1	Transaction Details Module
Functional Requirements 2.1	1) Used to display the transactions made by customers living in a given zip code for a given month and year. Order by day in descending order. 2) Used to display the number and total values of transactions for a given type. 3) Used to display the total number and total values of transactions for branches in a given state.

2.2 Customer Details Module

Req-2.2	Customer Details
Functional Requirements 2.2	1) Used to check the existing account details of a customer. 2) Used to modify the existing account details of a customer. 3) Used to generate a monthly bill for a credit card number for a given month and year. 4) Used to display the transactions made by a customer between two dates. Order by year, month, and day in descending order.

3. Functional Requirements - Data Analysis and Visualization

After data is loaded into the database, users can make changes from the front end, and they can also view data from the front end. Now, the business analyst team wants to analyze and visualize the data.

Use Python libraries for the below requirements:

Req - 3	Data Analysis and Visualization
Functional Requirements 3.1	Find and plot which transaction type has a high rate of transactions. Note: Take a screenshot of the graphs.
Functional Requirements 3.2	Find and plot which state has a high number of customers. Note: Take a screenshot of the graphs.
Functional Requirements 3.3	Find and plot the sum of all transactions for the top 10 customers, and which customer has the highest transaction amount. Hint (use CUST_SSN). Note: Take a screenshot of the graphs.

Overview of LOAN Application Data API

Banks deal in all home loans. They have a presence across all urban, semi-urban, and rural areas. Customers first apply for a home loan; after that, a company will validate the customer's eligibility for a loan.

Banks want to automate the loan eligibility process (in realtime) based on customer details provided while filling out the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. To automate this process, they have the task of identifying the customer segments to those who are eligible for loan amounts so that they can specifically target these customers. Here they have provided a partial dataset.

API Endpoint:

https://raw.githubusercontent.com/platformps/LoanDataset/main/loan_data.json

The above URL allows you to access information for loan application information. This dataset has all of the required fields for a loan application. You can access data from a REST API by sending an HTTP request and processing the response.

4. Functional Requirements - LOAN Application Dataset

Req-4	**Access to Loan API Endpoint**
Functional Requirements 4.1	Create a Python program to GET (consume) data from the above API endpoint for the loan application dataset.
Functional Requirements 4.2	Find the status code of the above API endpoint. Hint: status code could be 200, 400, 404, 401.
Functional Requirements 4.3	Once Python reads data from the API, utilize PySpark to load data into RDBMS (SQL). The table name should be **CDW-SAPP_loan_application** in the database. Note: Use the **"creditcard_capstone"** database**.**

5. Functional Requirements - Data Analysis and Visualization for LOAN Application

After the data is loaded into the database, the business analyst team wants to analyze and visualize the data.

Use Python libraries for the below requirements:

Req-5	**Data Analysis and Visualization**
Functional Requirements 5.1	Find and plot the percentage of applications approved for self-employed applicants. Note: Take a screenshot of the graph.
Functional Requirements 5.2	Find the percentage of rejection for married male applicants. Note: Take a screenshot of the graph.
Functional Requirements 5.3	Find and plot the top three months with the largest transaction data. Note: Take a screenshot of the graph.
Functional Requirements 5.4	Find and plot which branch processed the highest total dollar value of healthcare transactions. Note: Take a screenshot of the graph.

Capstone Project Submission Guidelines.

- Upload your project to your private GitHub repository, which needs to have a minimum of one separate branch off of the main branch.
- Virtual Environment with requirements.txt.
- Have a minimum of three commits (one-line brief overview, plus a more descriptive paragraph underneath, if needed).
- The project needs to have a “readme” file documenting the project details, information, and technical challenges, along with how they were resolved.
- Ensure that any sensitive information (*secret.text* files etc.) has been .gitignore.
- Upload all Python codes, PySpark codes, database scripts, and databases that are part of the repo.
- Take a screenshot of all of the graphs.
- Submit the GitHub repository link during the Canvas assignment submission.
- All learners should present their projects individually. Your instructor will schedule the presentation slot for you.

References:

PySpark:
https://spark.apache.org/docs/latest/api/python/index.html

Apache Spark - Spark SQL:
https://spark.apache.org/sql/

Analyzing and Visualization:

https://www.analyticsvidhya.com/blog/2021/08/understanding-bar-plots-in-python-beginners-guide-to-data-visualization/