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.

Instructions for Python Environment

You can work in a notebook while developing your project, but you must export it to a **.py** file that works appropriately. We ***expect*** your front-end user interface program to be fully functional when you demonstrate your final project.

Capstone Presentation Guide for Learners

Please visit the following document for detailed guidelines on the Capstone presentation and demonstration:
Capstone Presentation Guide for Learners (Data Engineering)

Capstone Project Submission Guidelines.

1. Submit a zipped copy of your entire capstone repository and all related files to the Canvas assignment submission.
    - Upload all Python codes (Jupyter notebook or VS notebook), PySpark codes, database scripts, and databases that are part of the project.
    - Take a screenshot of all of the graphs.

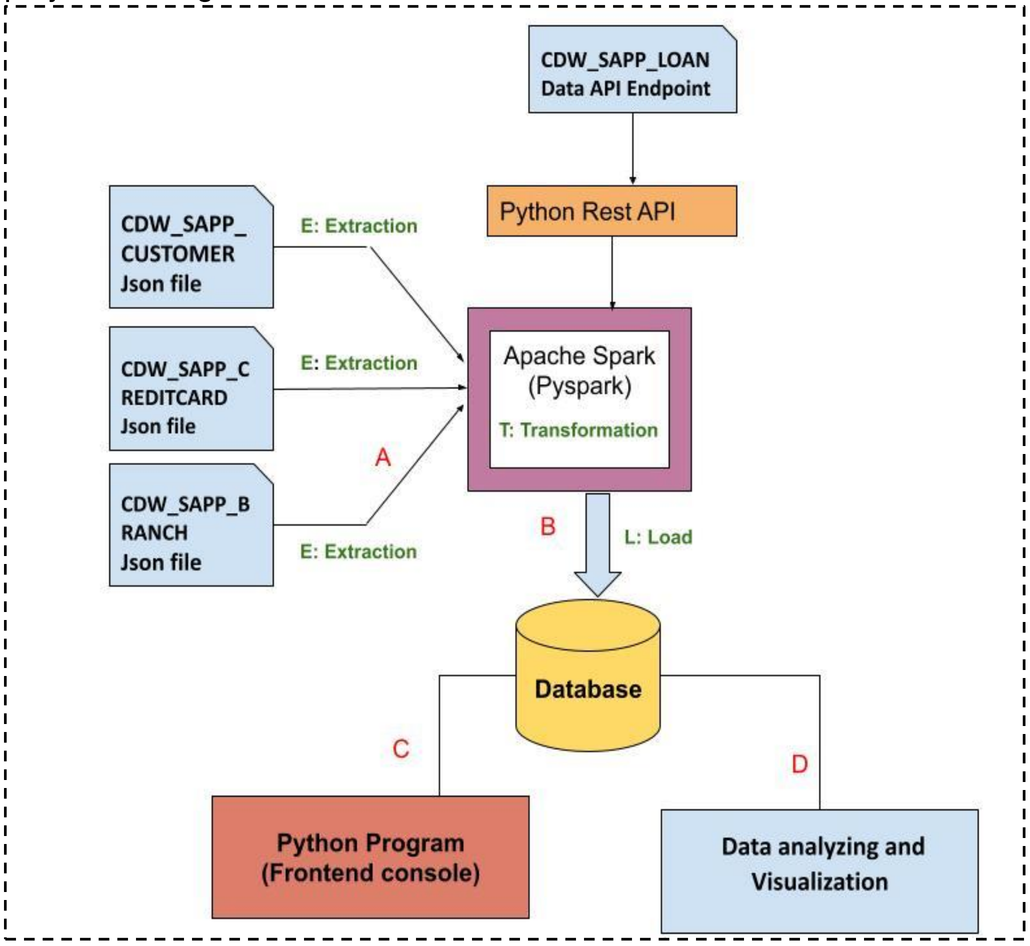
2. Submit on GitHub:
   - Upload your capstone project to your private GitHub repository, which needs to have a minimum of one separate branch off of the main branch.
     - Ensure that any sensitive information (*secret.text* files etc.) has been .gitignore.
     - Upload all Python codes, PySpark codes, database scripts, and databases that are part of the repo.
     - Take a screenshot of all of the graphs.
     - Create a `readme` file in your GitHub repository that will be utilized as the "Report" requirement.
     - Submit the GitHub repository link to the Canvas assignment submission.
3. Perform a technical walkthrough of your project, focusing on the following components (it should not take more than 30 minutes):
   a. Technical Requirements
      i. The user interface of your front-end program.
      ii. Database along with clean data.
      iii. Your visualizations.
      iv. All functionalities and features of your solution

   b. Highlight how your application works from the technical perspective (high level)
   c. Be prepared to see additional visitors (potential employer(s) and Per Scholas leaders) during your technical walkthrough of the capstone project.

Be prepared to have a discussion and Q&A regarding your capstone project.

Workflow Diagram of the Requirements.

The workflow diagram below will help you visualize the flow and scope of this capstone project at a high level.


CDW_SAPP_LOAN Data API Endpoint
Python Rest API
CDW_SAPP_CUSTOMER Json file
E: Extraction
CDW_SAPP_CREDITCARD Json file
E: Extraction
CDW_SAPP_BRANCH Json file
E: Extraction
A
Apache Spark (Pyspark)
T: Transformation
B
L: Load
Database
C
D
Python Program (Frontend console)
Data analyzing and Visualization

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	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 with 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/text menu**) to see/display data. For that, create a **console-based Menu (Python program)** to satisfy Functional Requirements 2 (2.1 and 2.2).

Here is a good walkthrough on what a console-based program looks like: https://www.geeksforgeeks.org/how-to-make-a-todo-list-cli-application-using-python/

You must be able to run it from a console.

*Note: You can work in a notebook while developing your project, but you must export it to a .py file that works appropriately. We **expect** your front-end user interface program to be fully functional when you demonstrate your final project.*

2.1 Transaction Details Module

Req-2.1	Transaction Details Module
Functional Requirements 2.1	1) Create a function that accomplishes the following tasks: **2.1.1 -** Prompt the user for a zip code, provide contextual cues for valid input, and verify it is in the correct format. **2.1.2 -** Ask for a month and year, and provide contextual cues for valid input and verify it is in the correct format. **2.1.3 -** Use the provided inputs to query the database and retrieve a list of transactions made by customers in the specified zip code for the given month and year. **2.1.4 -** Sort the transactions by day in descending order. **Remember**: this function should be callable from the main application interface and the output should be screen-reading friendly for the user.

2.2 Customer Details Module

Req - 2.2	Customer Details
Functional Requirements 2.2	**2.2.1.** Used to check the existing account details of a customer. **2.2.2.** Used to modify the existing account details of a customer. **2.2.3.** Used to generate a monthly bill for a credit card number for a given month and year. **2.2.4.** 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	*Create an appropriate visualization to perform the following task:* ● **Calculate and plot which transaction type has the highest transaction count.** *Note: Take a screenshot of the graph. Save a copy of the visualization, making sure it is PROPERLY NAMED!*
Functional Requirements 3.2	*Create an appropriate visualization to perform the following task:* ● **Calculate and plot the top 10 states with the highest number of customers.** *Note: Take a screenshot of the graph. Save a copy of the visualization, making sure it is PROPERLY NAMED!*
Functional Requirements 3.3	*Create a single appropriate visualization to perform the following task:* ● **Calculate the total transaction sum for each customer based on their individual transactions. Identify the top 10 customers with the highest transaction amounts (in dollar value). Create a plot to showcase these top customers and their transaction sums**.

	Hint (use CUST_SSN). *Note: Take a screenshot of the graph. Save a copy of the visualization, making sure it is PROPERLY NAMED!*

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 real time) 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 are tasked with identifying the customer segments of 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 related to loan applications. 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	Calculate the status code of the above API endpoint. Hint: status codes could be 200, 400, 404, or 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	*Create an appropriate visualization to perform the following task:* ● Calculate and plot the percentage of applications approved for self-employed applicants. Use the appropriate chart or graph to represent this data. *Note: Take a screenshot of the graph. Save a copy of the visualization, making sure it is PROPERLY NAMED!*
Functional Requirements 5.2	**Objective:** You will independently explore the provided **Credit Card** and **Loan Application** datasets to discover **actionable insights**. Your task is to design and answer your own data-driven question(s) and present your findings using effective visualizations. **What You Will Do:** 1. **Formulate Your Own Prompt** ○ Think like a data analyst or data engineer. ○ Define a question or hypothesis you want to explore using the given dataset. ○ Your prompt should aim to reveal a pattern, support a decision, or highlight an important finding. 2. **Analyze the Dataset** ○ Use **Pandas**, **PySpark**, or **SQL** to query, clean, and analyze the data. ○ Focus on creating **meaningful groupings, aggregations, or comparisons**. ○ Think about time periods, locations, user behavior, or transaction types.

	3. **Create Visualizations** ○ Develop at least **three** different types of charts or graphs. ○ Your visualizations should clearly communicate your insight. ○ Use appropriate chart types (bar, line, scatter, heatmap, etc.). ○ Add titles, axis labels, and legends where applicable. 4. **Explain Your Insight** ○ In your presentation, please include a brief explanation of the following topics: ■ What question did you ask? ■ What does your graph show? ■ What could someone (e.g., a manager or stakeholder) do with this information? 5. **Integrate in Capstone Presentation** ○ Include a visualization as part of your final capstone demo. ○ Be prepared to describe your reasoning, analysis steps, and implications. *Note: Take a screenshot of the graph. Save a copy of the visualization, making sure it is PROPERLY NAMED!*

6. Technical Walkthrough and Report

See above for instructions on the deliverables.

Non-functional Requirements - 6.1: Submission	Submit a zipped copy of your entire capstone repository and all related files to the Canvas assignment. ● Upload all Python codes (Jupyter notebook or VS notebook), PySpark codes, database scripts, and databases that are part of the project. ● Upload Screenshots for Graphs
Non-functional Requirements - 6.2: Walkthrough and Report	Perform a walkthrough of your project, focusing on the following components (it should not take more than 30 minutes): ● GitHub Repository ● The user interface of your front-end program. ● Database along with clean data. ● Your Graphs and visualizations ● Report (either in GitHub readme or Google Doc)

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/