5th Floor,
One Padmore Place,
off George Padmore
Road. Nairobi, Kenya
P.O. Box 5980-00200.

Savannah Informatics -Data Engineering
Nairobi, Kenya - Full time
Position: Junior - Mid Level hire

Data Engineering Screening Challenge

This assessment is designed to evaluate your ability to design, implement, and document a complete ETL pipeline using real-world data. The task mimics challenges faced in a data engineering environment, involving data ingestion, cleaning, transformation, and analysis using modern cloud tools.

Objective

Build a pipeline that processes claims and user data by:

1. Extracting data from multiple APIs.
2. Cleaning and normalizing the data.
3. Joining datasets to generate insights.
4. Automating the pipeline and documenting your work.

We encourage you to use Apache Airflow, but you may use normal python scripts if preferred. Follow software engineering best practices, including modular code, clear documentation, and adherence to coding standards.l

Project Details

Tech Stack

1. **Primary Tools**: Python, Google BigQuery.
2. **Free BigQuery Tier**: Google Cloud Free Tier provides 50 GB of free storage and 1 TB of free queries per month. Create a free Google Cloud account to access BigQuery and GCS. **Optionally if you are not able to get a Google Cloud Account you can save the data in your repository as csv files but bigquery is encouraged**.

Tasks

1. Extract Data from APIs

Use the following public APIs to retrieve data. Save the raw JSON data in Google Cloud Storage (GCS):

1. Users Data
2. Products Data
3. Carts Data

Example: Use requests or any HTTP library in Python to fetch data.

2. Clean and Normalize Data

Prepare the following cleaned datasets:

- Users Table:
  - Fields: user_id, first_name, last_name, gender, age, street, city, postal_code.
  - Tasks: Extract and flatten the address field into street, city, and postal_code.
- Products Table:
  - Fields: product_id, name, category, brand, price.
  - Tasks: Exclude products with price <= 50.
- Carts Table:
  - Fields: cart_id, user_id, product_id, quantity, price, total_cart_value.
  - Tasks: Flatten the products array into one row per product. Add total_cart_value for each cart.

3. Load Data into BigQuery

Store the cleaned datasets in separate BigQuery tables:

1. users_table
2. products_table
3. Carts_table

4. Join and Enrich Data

Perform the following joins in BigQuery:

- Users and Carts:
  - Combine demographic data with transaction details.
- Carts and Products:
  - Enrich transaction data with product details.

Generate the following datasets:

- User Summary:
  - Fields: user_id, first_name, total_spent, total_items, age, city.
  - Insights: Total spending and number of purchases per user.
- Category Summary:

  - Fields: category, total_sales, total_items_sold.
  - Insights: Aggregate sales by product category.
- Cart Details:
  - Fields: cart_id, user_id, product_id, quantity, price, total_cart_value.
  - Insights: Transaction-level details enriched with user and product data.

5. Automate with Orchestration

Automate the pipeline. Use Apache Airflow or python to orchestrate/run the workflow. Essentially we want to see these tasks run in this order:

- Extract data from APIs.
- Save raw JSON locally.
- Clean and load data into BigQuery.
- Run transformations and generate outputs.

6. Document Your Work

Provide clear documentation:

- Pipeline Design: Include DAG structure and task dependencies.
- Codebase Overview: Explain your scripts and modules.
- BigQuery Queries: Share and explain your SQL logic.
- Assumptions and Trade-offs: Highlight decisions made during implementation.

Expectations and Best Practices

We value the following:

1. Modular Code: Break down functionality into reusable functions/modules.
2. Error Handling: Manage API failures, missing data, or invalid inputs gracefully.
3. Scalability: Design workflows that can handle larger datasets in the future.
4. Version Control: Use Git to manage your code. Include a clear README.md.
5. Clarity: Ensure code and documentation are easy to understand.

Resources

1. Google Cloud Free Tier
2. Dummy JSON API
3. BigQuery Documentation
4. Apache Airflow Documentation

Submission

Please submit the following:

1. Code Repository: A GitHub or GitLab repository with all code and documentation.
2. BigQuery Links: Shared BigQuery tables or query results.
3. Presentation/Document: A brief write-up or slide deck summarizing your approach and findings.

We're excited to see your solution! If you have any questions during the assessment period, feel free to reach out. Good luck!