Narravance

FULL-STACK ENGINEER: SCREENING TASK

Task Overview

You are to create a data sourcing and visualization web app. A user should be able to create a new task of pulling data from at least two external sources, and then be able to see some analytics on that data once the task is complete. For example, you could use these external sources:

- Source A: A hosted/local JSON file providing a list of records (e.g., sales records of a car dealer)
- Source B: A CSV file (either hosted online or included in your repository) with similar records. (e.g., sales records of another car dealer)

Both sources should have a few overlapping fields (e.g., company, car model, date of sale, and price) to merge the data into a single unified table. We will use this example in the rest of the document to help you understand the assignment better. Feel free to use some other data format or problem (this shows your creativity :)).

Detailed Requirements

Task creation frontend

- The user can create a new task from the frontend by putting in some filtering parameters for their external source data. E.g.: Request sales records from 2023 to 2025 from Source A and Source B. In case of Source B, only restrict to Honda and Toyota cars.

Job queue

- Implement a simple in-memory job queue (e.g., using Python's queue module or a list) to simulate task processing for getting this data from the external sources.
- When a task is submitted, it starts in "pending" status. After a simulated delay (e.g., 5-10 seconds), it moves to "in progress", and after another delay, it becomes "completed". Update the database with the data received from the external data source(s).

Database

- Design a unified relational schema (using SQLite or any lightweight DB) that stores the data rows retrieved from that task. Make sure all rows are associated with a common task ID.
- Use a Python ORM (e.g., SQLAlchemy) for database interactions.

Visualization of retrieved task data on the frontend

Once the task is complete, the user should be able to see some analytics on the data that was pulled for their task.

- Use a visualization library (preferably D3.js) to display at least two interactive charts (e.g., a line chart for time series of rows v/s year, a bar chart for aggregated sales by car companies)

- Implement filtering options that update the charts dynamically (e.g., see this analytics for only 2024 data, see this analytics for only Honda rows)

Submission Instructions:

- Set up a repository on GitHub and upload all the necessary files there
- Record a demo of a walkthrough of the system and set it up as an unlisted YouTube video

Send your **resume, repository, and the video link** in an email to pratik@narravance.ai

Evaluation Criteria

- Architecture and Design
    - How well are the data ingestion, storage, and job management components designed?
    - The effectiveness of the API endpoints (speed, security, clarity).

- Frontend and Visualization
    - The interactivity, clarity, and visual appeal of the data visualizations and the overall application.
    - UX of the web application: Creating a task, checking previous tasks, and checking analytics on one particular task

- Code Quality & Documentation
    - Readability, modularity, and maintainability of the code
    - Completeness and clarity of the documentation.

Estimated Time:

The estimated time to complete this assignment is 2-3 days, depending on your proficiency and the complexity of the features you choose to implement.

Additional Notes:

- You are free to choose any additional features that showcase your skills and creativity.
- Feel free to reach out to pratik@narravance.ai if you have any questions or require clarification on the assignment.

Good luck!

As a reminder, here is the job description: https://tinyurl.com/nvc-fs-jd