Full Stack Engineer Technical Exercise

Overview

This technical exercise is designed to assess your skills as a full stack engineer, focusing on your ability to create a backend service that processes external API data and a frontend web application that interacts with this backend to present meaningful insights through data visualization.

Exercise 1: Backend Development

Objective

The task is to develop a backend service that retrieves data from the SpaceX API, processes it, and exposes it through a RESTful API.

Requirements

1. API Integration
    - Integrate with the SpaceX API to retrieve data for:
        - Rockets
        - Launches
        - Starlink satellites
    - Implement efficient data fetching strategies, such as caching or scheduled updates, to minimize calls to the SpaceX API.
2. Data Processing
    - Process and aggregate the raw data from the SpaceX API to create meaningful summaries and statistics.
    - Implement any necessary data transformations to support the frontend visualizations.
3. RESTful API Development
    - Create a RESTful API with the following endpoints:
        - `/api/dashboard`: Returns summary statistics and key metrics for the dashboard view.
        - `/api/rockets`: Returns processed data about SpaceX rockets.
        - `/api/launches`: Returns processed data about SpaceX launches.
        - `/api/starlink`: Returns processed data about Starlink satellites.

    - Implement proper error handling and status codes for all endpoints.
    - Add basic filtering, sorting, and pagination capabilities to the API endpoints where appropriate.
4. Documentation
    - Provide clear documentation for your API endpoints, including request/response formats and any query parameters.

Exercise 2: Frontend Development

Objective

Building on the backend service created in Exercise 1, develop a frontend web application that retrieves data from your custom API and visualizes it effectively.

Requirements

1. Frontend Development
    - Develop a frontend application with 3 views using a **Python** modern frontend framework or library of your choice (e.g., Vizro, Taipy).
    - Create the following views:
        1. Dashboard: An overview page with summary statistics and key metrics.
        2. Rockets and Launches: A page dedicated to visualizing data about SpaceX rockets and launches.
        3. Starlink: A page focusing on Starlink satellite data and orbital information.
2. API Integration
    - Integrate with your custom backend API to retrieve the processed SpaceX data.
    - Implement efficient data fetching strategies, such as caching or state management, to optimize frontend performance.
3. Data Visualization
    - Utilize a charting library of your choice to create interactive and informative visualizations.
    - Implement at least one visualization for each of the following datasets:
        1. Rockets: Compare specifications or success rates of different rocket models.
        2. Launches: Visualize launch frequency over time or success rates by year.

        3. Starlink: Display satellite positions on a world map or visualize orbital parameters.
4. Additional Features
    - Implement error handling and loading states for API requests.
    - Add filtering or sorting capabilities for the data presented, utilizing the backend API's functionality.
    - Include a search functionality for finding specific rockets, launches, or satellites.

SpaceX Data API Docs

https://github.com/r-spacex/SpaceX-API/tree/master/docs