Coding Challenge: Customer Behavior Analytics Tool

Task Overview

Develop an application that processes a stream of customer activity events from a local file, generates insights into customer behavior, and exposes these insights through a REST API.

Scenario

The application will simulate real-time processing of customer activity on an e-commerce platform to identify trends and insights such as popular products, customer engagement, and sales performance.

Objectives

Event Stream Simulation: Create a local file (e.g., `events.jsonl`) containing simulated event data in a JSONL format with events like `page_visit`, `add_to_cart`, and `purchase`. Each event should include attributes such as `customer_id`, `timestamp`, `item_id`, and any other relevant information.

Event Processing:

- Implement logic to read and process these events from the file as if it were a live stream.
- Analyze the events to calculate metrics such as:
    - The most viewed item in the last hour.
    - The item most added to cart in the last 24 hours.
    - The highest-grossing item (total sales) in the last 24 hours.

REST API: Develop a REST API to expose the insights generated by the application. The API should support queries to retrieve:

- The current most viewed item.
- The current top item added to cart.
- The current highest-grossing item.

Testing: Please include a set of basic test cases to demonstrate that the application works as expected.

Documentation: Provide a README file with:

- Instructions on how to set up and run the application.
- A description of the API endpoints and how to use them.
- An overview of the event processing logic and assumptions made.

Considerations

- Focus on clean, readable, and efficient code.
- Use appropriate data structures and algorithms to ensure that the application can process events and calculate metrics in a performant manner.
- Consider error handling, especially for reading and processing the event stream.

Deliverables

- Source code for the application, including any configuration or supporting files.
- A sample `events.jsonl` file with simulated event data for testing.
- A README file with comprehensive setup, run instructions, and API documentation.
- Optional Challenge: Data Aging and Historical Insights - Devise a mechanism for aging data to provide long-term insights. As events move beyond the 24-hour real-time analysis window, aggregate them into broader time buckets (e.g., daily, weekly, monthly) to enable historical trend analysis without overwhelming memory or storage. This approach should balance detail and performance, offering a scalable way to retain and query long-term data trends.