General TODOS.  
  
These are TODOs which have been skipped, still need to get done.

Contents

[**General TODOS.**  These are TODOs which have been skipped, still need to get done. 1](#_Toc186194669)

[SPARK DATA PROCESSING 1](#_Toc186194670)

[**User Behavior Analysis (Optional)** 1](#_Toc186194671)

[**Data Cleanup and Enrichment (Optional)** 2](#_Toc186194672)

[**Full ETL Pipeline Automation (Optional)** 2](#_Toc186194673)

[**Next Steps:** 2](#_Toc186194674)

[JAVA BACKEND 3](#_Toc186194675)

[Next Steps: 3](#_Toc186194676)

[Postman 3](#_Toc186194677)

[Write Unit and Integration Tests 4](#_Toc186194678)

[Document Your API 4](#_Toc186194679)

# SPARK DATA PROCESSING

**User Behavior Analysis (Optional)**

* **Objective**: Analyze user behavior logs to uncover insights such as popular search routes, frequent search times, or booking conversion rates.
* **Input Data**:
  + **User Logs**: user\_logs.csv
* **Processing Steps**:
  + **Parse User Data**:
    - Analyze user\_logs.csv to extract key metrics such as:
      * Most frequently searched routes.
      * Times of day with peak searches.
      * Conversion rates (percentage of searches resulting in bookings).
  + **Group and Aggregate**:
    - Perform grouping and aggregation using Spark SQL or PySpark DataFrame operations.
  + **Output**:
    - Create one or more tables with metrics, e.g.:

origin, destination, search\_count, booking\_rate

* **Output**: Write insights to PostgreSQL or a CSV in S3.

**Data Cleanup and Enrichment (Optional)**

* **Objective**: Perform data cleaning and enrichment before running other jobs.
* **Processing Steps**:
  1. **Handle Missing Data**:
     + Fill or drop missing values in critical fields such as price, availability, and booking status.
  2. **Data Enrichment**:
     + Add derived fields (e.g., calculate days\_to\_departure from departure\_date).
* **Output**: Write the cleaned and enriched data to intermediate tables in S3 or PostgreSQL.

**Full ETL Pipeline Automation (Optional)**

* **Objective**: Combine multiple Spark jobs into a single pipeline for end-to-end processing.
* **Processing Steps**:
  1. Start with **data cleaning and enrichment**.
  2. Perform **Audience Segmentation**.
  3. Execute **Alternative Dates** analysis.
  4. Generate **Ad Recommendations**.
* **Output**: Final tables or datasets ready for API consumption.

**Next Steps:**

* **Focus on the Core Jobs First**:
  1. **Alternative Dates**
  2. **Ad Recommendations**
* **Set Up Airflow Integration**:
  1. Define these Spark jobs as steps in your Airflow DAG for orchestration.
* **Test Results**:
  1. Save outputs to S3 or PostgreSQL and validate correctness before integrating with APIs.

Let me know if you'd like a detailed outline or implementation plan for any specific job!

# JAVA BACKEND

Unit Tests, Documentation, Testing with Postman

### Next Steps:

Postman**:**

* + Start the Spring Boot application and ensure it initializes correctly.
  + Test the gRPC service locally using tools like Postman for gRPC, grpcurl, or a small client written in Java.

Write Unit and Integration Tests**:**

* + Add tests for your FlightFlexServiceImpl and repositories.
  + Include both happy-path and edge-case scenarios.

Document Your API**:**

* + Ensure your .proto file is well-documented.
  + Provide clear instructions on how clients can interact with the gRPC API.

**Frontend Integration:**

* + If the gRPC API serves as a backend for a React frontend, plan how your frontend will interact with it, possibly using a gRPC-Web proxy.