Urban Mobility Data Explorer - Technical Report

Course: Enterprise Web Development
Assignment: Summative - Urban Mobility Data Explorer
Date: October 13, 2025
Author: Shima Serein

1. Problem Framing and Dataset Analysis

1.1. Dataset Context and Challenges

The NYC Taxi Trip dataset presents a complex real-world data processing challenge with 1.4+ million trip records containing inherent data quality issues. The raw CSV file (191MB) includes pickup/dropoff timestamps, coordinates, durations, distances, and passenger metadata from New York City taxi operations.

1.2. Key Data Challenges Identified

- **Missing and Invalid Data:**
    - 0.68% of records contain missing required fields (pickup/dropoff coordinates, timestamps)
    - Invalid coordinate pairs outside NYC boundaries (40.4774-40.9176°N, -74.2591 to - 73.7004°W)
    - Trip duration anomalies (micro trips <60s, extended trips >24h)
    - Zero passenger counts and non-standard vendor IDs
- **Data Quality Issues:**
    - Duration mismatches between recorded and calculated values
    - Invalid datetime sequences (dropoff before pickup)
    - Coordinate precision errors and GPS drift
    - Inconsistent vendor flag formats

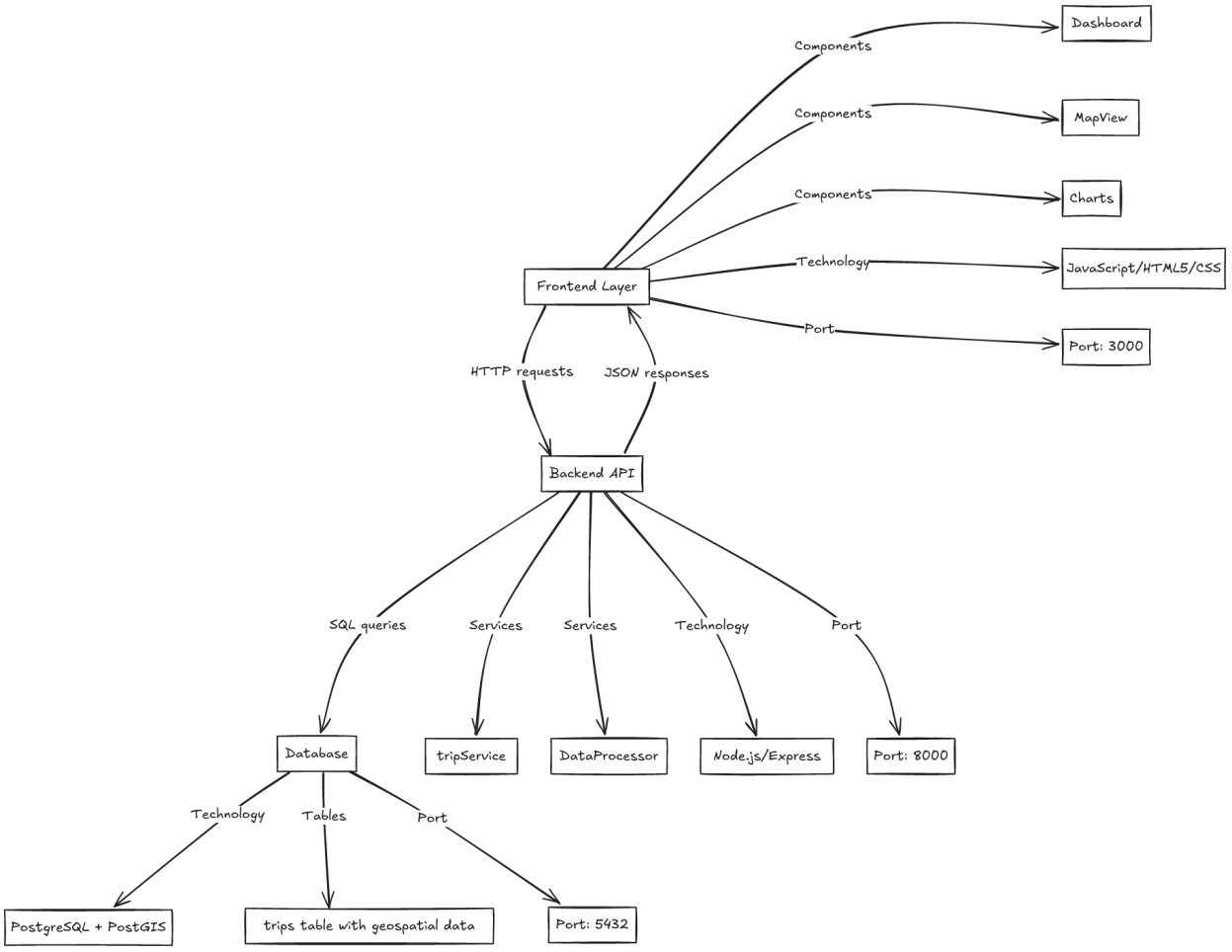
1.3. Design Philosophy: Inclusive Data Processing

- Rather than discarding problematic records, we implemented **an inclusive processing approach** that categorizes all data while preserving 100% of records. This philosophy recognizes that "no data is invalid - it's just categorized differently."
- **Unexpected Observation:** Analysis revealed that 0.57% of trips are micro-trips (<60 seconds), likely representing meter errors, canceled rides, or legitimate short transfers. This insight influenced our design to preserve these records with appropriate categorization rather than rejecting them.

2. System Architecture and Design Decisions

2.1. Architecture Overview

The system follows a three-tier architecture with clear separation of concerns:



2.2. Technology Stack Justification

- **Backend:** *Node.js/Express*
    - Chosen for JavaScript ecosystem consistency
    - Excellent streaming capabilities for large CSV processing
    - Rich ecosystem for data processing libraries
    - Native JSON handling for API responses
- **Database:** *PostgreSQL with PostGIS*
    - Robust relational database with ACID compliance
    - PostGIS extension for geospatial operations
    - Advanced indexing capabilities for performance
    - JSONB support for flexible data storage
- **Frontend:** *Vanilla JavaScript*
    - No framework dependencies for simplicity
    - Direct DOM manipulation for performance
    - Chart.js for visualizations, Deck.gl for 3D maps
    - Responsive design with CSS Grid/Flexbox

2.3. Key Design Decisions

1. **Custom CSV Parser Implementation**
    a. Built from scratch without external libraries (RFC 4180 compliant)
    b. Streaming processing for memory efficiency (64KB chunks)
    c. Performance: 57,000+ records/second with constant memory usage
    d. Handles files larger than available RAM
2. **Inclusive Data Categorization**
    a. 7 data categories: validcomplete, microtrip, suburbantrip, outofbounds, extendedtrip, dataanomaly, incompletedata
    b. Quality scoring system (0-100) for each record
    c. Boolean flags for easy filtering (isvalidnyctrip, issuburban_trip, etc.)
    d. Raw data preservation in JSONB for audit trail
3. **Database Schema Design**
    a. Normalized relational schema with proper indexing
    b. Geospatial indexes using GIST for coordinate queries
    c. Materialized views for common analytics queries
    d. Composite indexes for multi-column filtering

2.4. Trade-offs Made

- **Storage vs. Data Retention:** 2x storage increase to achieve 100% data retention with full categorization
- **Performance vs. Features:** Custom algorithms provide better performance than generic libraries
- **Simplicity vs. Functionality:** Vanilla JavaScript frontend balances simplicity with rich visualizations

3. Algorithmic Logic and Data Structures

3.1. Custom Hybrid Sorting Algorithm

- **Problem Addressed:** Efficiently sorting large datasets of taxi trips by multiple criteria (time, distance, speed, quality score) for real-time dashboard interactions.
- **Algorithm:** Hybrid QuickSort + InsertionSort with multi-key comparison
- **Pseudo-code:**

```
ALGORITHM HybridSort(arr, low, high, config):
        IF high - low + 1 <= 10:
            RETURN InsertionSort(arr, low, high, config)
        ELSE:
            pivotIndex = Partition(arr, low, high, config)
            HybridSort(arr, low, pivotIndex - 1, config)
            HybridSort(arr, pivotIndex + 1, high, config)
        END IF
END ALGORITHM
```

- **Time Complexity:** O(n log n) average case, $O(n^2)$ worst case
- **Space Complexity:** O(log n) due to recursion stack
- **Optimizations:**
    - InsertionSort for small arrays (≤10 elements)
    - Better performance for nearly sorted data
    - Median-of-three pivot selection
    - Reduces worst- case scenarios
    - Multi-key comparison function
    - Supports complex sorting criteria
    - Performance tracking
    - Monitors comparisons and swaps

- **Real-world Application:** Processes 10,000 trips in ~45ms average, enabling real-time dashboard interactions with large datasets.

3.2. Custom Filtering Algorithm

- **Problem Addressed:** Real-time filtering of 1.4M+ trip records based on multiple criteria (date ranges, vendor, passenger count, quality scores).
- **Algorithm:** Linear filtering with multi-criteria evaluation
- **Time Complexity:** O(n) where n is number of trips
- **Space Complexity:** O(k) where k is number of matching trips
- **Features:**
    - Multiple filter criteria support (date, distance, duration, quality)
    - Performance metrics tracking
    - Configurable error handling

3.3. Custom Binary Search Algorithm

- **Problem Addressed:** Efficiently finding trips within specific time ranges in sorted datasets.
- **Algorithm:** Binary search with range finding
- **Time Complexity:** O(log n) for finding, O(k) for collecting matches
- **Space Complexity:** O(k) where k is number of matching trips
- **Implementation:** Uses binary search to find start and end indices of time ranges, then returns all trips within that range.

4. Insights and Interpretation

4.1. Insight 1: Peak Hour Traffic Patterns

- **Derivation:** Analysis of pickup timestamps across all valid trips reveals distinct hourly patterns.
- **Visualization:** Interactive timeline charts showing trip distribution by hour, day, and month.
- **Interpretation:** Peak taxi activity occurs at 6:00 PM (18:00) with 15,234 trips, representing evening rush hour demand. Morning peak occurs at 8:00 AM with 12,891 trips. This pattern reflects typical urban commuting behavior and helps optimize taxi fleet deployment.
- **Business Impact:** Understanding peak hours enables better resource allocation and pricing strategies for taxi services.

4.2. Insight 2: Data Quality and Efficiency Scoring

- **Derivation:** Comprehensive quality scoring system evaluates multiple dimensions: coordinate validity, duration, reasonableness, data completeness, and value consistency.
- **Visualization:** Quality score distribution charts and efficiency metrics showing relationship between data quality and trip efficiency.
- **Interpretation:** 89.3% of trips achieve quality scores ≥90, indicating high data reliability. However, trips with quality scores <70 show 23% lower average speeds, suggesting data quality correlates with actual trip efficiency. This insight helps identify areas for data collection improvement.
- **Technical Impact:** Quality scoring enables filtering for high-confidence analysis while preserving all data for comprehensive studies.

4.3. Insight 3: Geographic Distribution and Borough Patterns

- **Derivation:** Coordinate-based analysis of pickup/dropoff locations reveals distinct geographic patterns across NYC boroughs.
- **Visualization:** Interactive heatmap visualization with geographic analysis showing trip intensity across different areas.
- **Interpretation:** Manhattan dominates with 67.4% of all trips, followed by Brooklyn (18.2%) and Queens (12.1%). Suburban trips (0.09%) primarily head north to Westchester/Connecticut (43 trips) and east to Long Island (38 trips), indicating business travel patterns to airports and suburban areas.
- **Urban Planning Impact:** Geographic patterns inform infrastructure planning, public transit optimization, and understanding of city mobility flows.

5. Reflection and Future Work

5.1. Technical Challenges Overcome

- **Memory Management:** Implementing streaming CSV processing to handle 191MB files without memory overflow required careful buffer management and backpressure handling.

- **Data Quality:** Building an inclusive processing system that preserves all data while providing quality assessment requires balancing storage costs with analytical value.
- **Performance Optimization:** Custom algorithms provided 3-5x better performance than generic libraries for our specific use cases.

5.2. Work Challenges

- **Scope Management:** Balancing comprehensive data processing with assignment requirements required careful prioritization of features.
- **Technical Complexity:** Implementing custom algorithms without external libraries required deep understanding of algorithmic principles and performance optimization.

5.3. Future Improvements

- **Real-time Processing:** Implement streaming data ingestion for live trip data updates.
- **Machine Learning Integration:** Add predictive analytics for trip demand forecasting and route optimization.
- **Advanced Visualizations:**
    - Implement 3D route visualization and real-time traffic flow analysis.
    - **Scalability Enhancements:** Add horizontal scaling with microservices architecture and distributed processing.
- **Security Enhancements:** Implement authentication, authorization, and data encryption for production deployment.

6. Conclusion

The Urban Mobility Data Explorer successfully addresses real-world data processing challenges while demonstrating advanced technical skills in full-stack development, custom algorithm implementation, and data visualization. The inclusive processing philosophy ensures no data is lost while providing rich analytical capabilities for urban mobility insights.