**Agent Performance Summary Generator - Project Report**

**Executive Summary**

The Agent Performance Summary Generator is a web-based application built with Streamlit that provides call center managers with a convenient tool to analyze agent performance metrics. The application processes data from three primary data sources (call logs, agent roster, and disposition summary) to generate comprehensive performance reports, highlighting key metrics such as connect rates, call volumes, and call durations. This tool serves as an essential management resource for evaluating agent productivity and identifying top performers.

**Project Overview**

**Purpose**

The application addresses the need for streamlined data processing and visualization in call center environments. It automates the transformation of raw call data into actionable insights, allowing managers to quickly identify performance trends and recognize top-performing agents.

**Target Users**

* Call center managers and supervisors
* Team leads responsible for agent performance
* Operations analysts tracking call center metrics

**Technical Architecture**

**Technology Stack**

* **Frontend & Backend:** Streamlit (Python-based framework)
* **Data Processing:** Pandas for data manipulation and analysis
* **Visualization:** Streamlit's native data visualization tools

**Data Flow**

1. User uploads three CSV files via the Streamlit interface
2. Application validates data structure and content
3. Data preprocessing and transformation occurs
4. Performance metrics are calculated and aggregated
5. Results are displayed in interactive tables
6. Users can download the processed data as CSV

**Key Functionalities**

**Data Validation and Preparation**

The application begins with robust data validation through the validate\_and\_prepare\_data() function, which:

* Verifies that required columns exist in each dataset
* Converts date columns to appropriate datetime format
* Validates numeric data types for calculations
* Returns error messages for missing or invalid data

**Data Processing**

The core logic in the process\_data() function:

* Merges the three datasets using agent\_id, org\_id, and call\_date as keys
* Creates derived features like 'presence' and 'completed'
* Aggregates data by agent and date
* Calculates performance metrics including:
  + Total calls handled
  + Number of unique loans processed
  + Call completion rate
  + Average call duration in minutes
  + Agent presence status

**Performance Reporting**

The application provides two types of reporting views:

1. **Comprehensive view** - A complete table of all agent performance metrics
2. **Summary views** - Focused reports highlighting top performers:
   * Top performer summary with key metrics
   * Top 3 performers comparison

**Data Export**

Users can download the processed performance data as a CSV file for further analysis or reporting.

**User Interface**

The application features a clean, intuitive interface with:

* A clear title and description of functionality
* File upload section with designated areas for each required file
* Success/error messaging for data validation
* Expandable sections to manage screen real estate
* Interactive tables for data visualization
* Download button for exporting results
* Toggle buttons to switch between different summary views

**Data Model**

**Input Data Requirements**

The application expects three CSV files with specific required columns:

1. **Call Logs**
   * Required: agent\_id, org\_id, call\_date
   * Optional: duration, call\_id, installment\_id, and others
2. **Agent Roster**
   * Required: agent\_id, org\_id
   * Contains: users\_first\_name, users\_last\_name, and other agent details
3. **Disposition Summary**
   * Required: agent\_id, org\_id, call\_date
   * Contains: login\_time, status, and other call outcome information

**Output Data Model**

The application generates an agent performance summary with the following key metrics:

* agent\_id
* call\_date
* users\_first\_name
* users\_last\_name
* total\_calls
* unique\_loans
* completed\_calls
* avg\_duration\_min
* presence
* connect\_rate

**Technical Implementation Details**

**Modular Design**

The code is structured into distinct functional components:

* Helper functions for data validation and processing
* Data transformation pipeline
* UI components and layout
* Report generation functions

**Error Handling**

The application includes error handling for:

* Missing required columns in input files
* Date format conversion errors
* Numeric data type validation

**Performance Considerations**

* Efficient data processing using Pandas operations
* Streamlined data merging and aggregation
* Minimal data transformations to ensure responsive performance

**Future Enhancements**

The following enhancements could be considered for future versions:

1. **Advanced Filtering Options**
   * Date range selection
   * Organization/team filtering
   * Custom metric thresholds
2. **Enhanced Visualizations**
   * Performance trend charts over time
   * Comparative analysis between agents
   * Customizable dashboards
3. **Additional Metrics**
   * Call quality scores
   * Customer satisfaction metrics
   * Performance against targets
4. **User Authentication**
   * Role-based access control
   * Personalized reports based on user role
5. **Automated Reporting**
   * Scheduled report generation
   * Email distribution of reports
   * Report archiving

**Conclusion**

The Agent Performance Summary Generator successfully addresses the need for efficient call center performance analysis by automating the process of transforming raw call data into actionable insights. The application's streamlined interface and robust data processing capabilities make it a valuable tool for call center management, enabling quick identification of performance trends and recognition of top performers.

The modular design and clear separation of concerns within the codebase allow for easy maintenance and future enhancements. With its focus on usability and relevant metrics, this application provides significant value to call center operations by reducing the time and effort required to generate performance reports.