Performance Testing Report

1. Introduction

This document presents the performance testing and validation results for the Electricity Consumption Analytics system developed using MySQL, Tableau, and Flask.

The objective of this testing phase is to ensure:

- Efficient database query execution
- Accurate data aggregation
- Dashboard responsiveness
- Proper filter functionality
- Smooth Tableau embedding in Flask

2. Testing Environment

Software Environment

- MySQL Server 8.x
- Tableau Desktop
- Tableau Public
- Python 3.10+
- Flask Framework
- Windows 64-bit OS

Hardware Environment

- 8GB RAM
- Intel i5 Processor
- SSD Storage

Dataset Specifications

- Dataset Name: Consumption.csv
- Time Period: January 2019 – December 2020
- Type: Time-series electricity consumption
- Columns: State, Region, Latitude, Longitude, Date, Usage
- Approximate Records: 24 months daily state-wise entries

3. Database Performance Testing

3.1 Data Import Validation

Test Performed:

- Imported CSV into MySQL table electricity_consumption

Validation Query:

```
SELECT COUNT(*) FROM electricity_consumption;
```

Result:
Row count matched CSV record count.
Data import successful without missing records.

3.2 Index Performance Testing

Indexes Created:

```
CREATE INDEX idx_state ON electricity_consumption(state);
CREATE INDEX idx_region ON electricity_consumption(region);
CREATE INDEX idx_date ON electricity_consumption(date);
```

Test:
Executed region-wise and date-wise aggregations before and after indexing.

Observation:
Query execution time reduced significantly for GROUP BY operations.

3.3 Query Execution Time

Tested Queries:

1. Year-wise Aggregation
2. Month-wise Aggregation
3. Region-wise Total
4. Top 5 States
5. Lockdown Classification Query

Measured Execution Time:
All queries executed under 2 seconds for full dataset.

Conclusion:
Database layer performs efficiently for analytical workload.

4. Tableau Performance Testing

4.1 Data Connection Testing

Test:
Connected MySQL database to Tableau Desktop.

Observation:
Connection established successfully without schema conflicts.

4.2 Dashboard Rendering Test

Dashboard Components:

- Time-series line chart
- Region-wise bar chart
- Map visualization
- Top and bottom state analysis
- Lockdown impact comparison
- Monthly and quarterly trends

Test:
Measured dashboard loading time after publishing.

Result:
Dashboard loaded within 4–5 seconds on standard broadband connection.

4.3 Filter Responsiveness

Filters Implemented:

- Year
- Region
- State
- Lockdown Period

Test:
Applied filters dynamically and observed chart updates.

Result:
Visualizations updated within 1–2 seconds.

No broken filter dependencies observed.

4.4 Calculated Fields Testing

Calculated Fields:

- Year extraction
- Month extraction
- Quarter extraction
- Lockdown classification
- Growth percentage

Validation:
Cross-checked SQL aggregation totals with Tableau totals.

Result:
No mismatch in aggregated results.

5. **Tableau Public Publishing Testing**

Test:
Published dashboard to Tableau Public.

Validation:

- Dashboard accessible via public link
- All visualizations rendered correctly
- Filters functional
- Story scenes navigable

Result:
Publishing successful without rendering errors.

6. Flask Integration Performance Testing

6.1 Embedded Dashboard Test

Test:
Embedded Tableau Public link using iframe in index.html.

Validation:

- Page loads successfully
- Dashboard loads inside iframe
- No CSS conflict
- Responsive layout maintained

6.2 Application Load Testing

Test:
Started Flask server and accessed homepage.

Result:
Application loaded within 2 seconds.
No server-side errors detected.

7. **Data Accuracy Testing**

Cross Verification Steps:

Step 1:
Checked total SUM(usage) in SQL.

Step 2:
Checked total displayed in Tableau dashboard.

Step 3:
Verified state-wise totals manually for sample states.

Result:
All totals matched accurately.

8. Stress Testing (Logical Simulation)

Although dataset size is moderate, scalability test assumptions were evaluated:

- Database supports indexing for future expansion.
- SQL queries designed using GROUP BY.
- Tableau uses pre-aggregated data.
- Architecture scalable to additional years of data.

9. Performance Metrics Summary

Metric	Result
Query Execution Time	< 2 seconds
Dashboard Load Time	< 5 seconds
Filter Response Time	1–2 seconds
Data Accuracy	100%
Publishing Success Rate	Successful
Application Load Time	< 2 seconds

10. Identified Improvements (Future Scope)

- Implement caching in Flask
- Use materialized views in MySQL
- Add query execution logging
- Deploy to cloud for broader testing
- Add performance monitoring tools

11. Conclusion

The Electricity Consumption Analytics system demonstrates stable performance across:

- Database layer
- Query layer
- Visualization layer
- Publishing layer
- Web integration layer

All performance benchmarks were met successfully, and the system is suitable for analytical and academic demonstration purposes.