

PhonePe Transaction Insights Dashboard — Final Project Report

This project, PhonePe Transaction Insights Dashboard, presents a complete data analytics pipeline — from extracting deeply nested JSON data to delivering a fully interactive, production-ready dashboard. The main objective was to convert unstructured raw data from the PhonePe Pulse GitHub repository into meaningful insights that can guide business and policy decisions around digital payment adoption in India.

The project includes multiple core components:

1. Data Extraction & Transformation:

Custom Python scripts were developed to extract hierarchical JSON files containing transaction, user, and insurance data. These were parsed, normalized, and transformed into tabular structures for easier analysis.

2. SQL-Based Modeling:

The structured data was stored in a relational SQLite database using a modular schema. This database supports both analytical workloads and dynamic queries based on state, district, year, quarter, and transaction types.

3. Exploratory Data Analysis (EDA):

Using Python libraries such as Pandas and Matplotlib/Plotly, multiple patterns were uncovered — such as:

- Rapid UPI adoption in Tier-2 and Tier-3 cities.
- Seasonal transaction spikes during festivals.
- Regional disparities in insurance product penetration.
- Transaction behavior changes during COVID-19 lockdown and recovery.
- Interactive Dashboard:
 - A Streamlit-based dashboard was created to visualize insights. The UI includes:
 - Dynamic filters (Year, Quarter, State, Transaction Type)
 - KPIs (Key Performance Indicators)
 - Animated charts and time-series plots
 - Choropleth maps using GeoJSON files for district-level comparisons

4. Frontend Deployment:

In addition to the backend dashboard, a React+Vite frontend was deployed via Netlify for an enhanced user experience. This frontend displays static visualizations, charts, and summary metrics for stakeholders.

5. Documentation & Testing:

- Full documentation is included (README, user guide, technical documentation)
- SQL queries and Python pipelines were tested using Pytest
- Folder structure adheres to industry best practices

Deliverables

- Python scripts for ETL, data cleaning, and visualization
- SQL schema files (schema.sql, schema_only.sql)
- Streamlit dashboard (streamlit_app.py)
- React frontend components (Netlify)
- Documentation (markdown files)
- Presentation slides with insights and recommendations

Guidelines Followed

- Clean, modular Python and SQL code
- Git version control for code collaboration and change tracking
- Pytest-based testing framework for quality assurance
- Markdown-based documentation for reproducibility and clarity

Key Outcomes

- Created an end-to-end data system from raw JSON to business insights
- Delivered a dashboard capable of analyzing transactions geographically and over time
- Identified fintech adoption trends across India
- Enabled decision-makers to explore real-world financial data visually and interactively

Future Enhancements

- Add predictive models (e.g., transaction forecasting by state)
- Migrate database from SQLite to PostgreSQL or Supabase for cloud scalability
- Integrate live APIs from PhonePe or similar sources for real-time updates
- Containerize the solution using Docker for production deployment
- Implement user roles and authentication for secure enterprise access

Conclusion

This project demonstrates how open-source data, when combined with the right tools and techniques, can provide powerful business intelligence. From data engineering to dashboard deployment, this full-stack project bridges raw data and decision-making. It offers valuable insights not just to data scientists and engineers but also to fintech strategists, marketers, and policy analysts working toward financial inclusion and digital adoption in India.