Stock Market Analysis Project

We will collect stock data from the Yahoo Finance API, clean and enrich it locally using Delta Lake and PySpark, and generate actionable business insights like price trends, volatility analysis, and technical indicators, all managed and automated locally.

Project Goal

- Ingest raw stock data from Yahoo Finance API
- Clean and transform the data (deduplication, imputation, validation)
- Compute financial metrics (e.g., RSI, Moving Averages, Volatility)
- Visualize insights with local dashboards
- Automate the pipeline end-to-end on a local machine

Phase 1: Data Ingestion (Bronze Layer)

Objective:
Collect raw stock data and store it locally in an efficient, queryable format.

Steps:
- Ingest stock data using yfinance
- Save as Delta Lake format (Parquet) on local disk
- Partition by symbol and date for performance
- Validate fields: timestamps, ticker formats, price values

Automation:
- Use Apache Airflow (locally) to schedule ingestion
- Monitor and log jobs with DAGs

Tech Used:
- yfinance, pandas
- Delta Lake (local)
- Apache Airflow (local)

Phase 2: Data Cleansing & Transformation (Silver Layer)

Objective:
Transform raw data into clean, enriched datasets ready for analytics.

Steps:
- De-duplicate records via Delta Lake MERGE
- Handle missing values (forward-fill/interpolation)

- Add technical indicators using pandas-ta or TA-Lib
- Remove invalid rows (NaNs, negatives)
- Optimize Delta Lake files (OPTIMIZE, VACUUM)

Tech Used:
- PySpark
- Delta Lake
- pandas-ta, pandas
- Great Expectations (optional)

Phase 3: Analytics & Insights (Gold Layer)

Objective:
Generate and analyze financial metrics and prepare for reporting.

Steps:
- Compute key metrics: RSI, MAs, Volatility, Sharpe Ratio
- Rolling stats: 5-day avg, 14-day RSI, etc.
- Use DBT (local) for SQL models
- Build local dashboards using Apache Superset

Tech Used:
- DBT (local)
- Apache Superset
- pandas, numpy
- Delta Lake

Phase 4: Automation & Reliability (Local)

Objective:
Ensure your local pipeline runs reliably and incrementally.

Steps:
- Use Airflow DAGs to trigger Silver after Bronze
- Enable incremental processing in Delta
- Use time travel in Delta Lake for rollback/debugging
- Optional: Archive old data manually

Tech Used:
- Apache Airflow
- Delta Lake
- Local file system

Summary

Layer	Purpose	Format	Tools
Bronze	Raw data ingestion	Delta Lake	yfinance, Airflow, Delta
Silver	Cleaned, enriched data	Delta Lake	PySpark, pandas-ta, Delta
Gold	Analytics & metrics	Delta Lake	DBT, Superset, pandas
Auto	Pipeline automation & checks	N/A	Airflow, Delta