**ETL PIPELINE FOR BUDGET DATA DASHBOARD**

**Objective**:

* **ETL Pipeline**: Extract data from uploaded Excel/CSV files, clean and transform (handle missing values, truncations, standardizations), and load into a database.
* **Dashboard**: Web-based interface for uploading data, viewing visualizations (e.g., trends, comparisons), and exporting results.
* **Data Handling**: Support large datasets (e.g., >1GB, millions of rows) with efficient processing.
* **Comparisons**: Enable side-by-side analysis of multiple datasets (e.g., different states or years).
* **Analysis**: Time series visuals (line charts for NSDP trends), compositions (pie charts for revenue sources), and comparisons (bar charts for Haryana vs. India).
* **Free of Cost**: Use only open-source tools and free hosting options.
* **Optional RAG**: Allow natural language queries (e.g., “Compare Haryana’s NSDP growth to India’s”).
* **RAG Capabilities**
* **Query Examples**:

"Compare education spending between Maharashtra and Gujarat"

"What are the key factors driving fiscal deficit in UP?"

"Show me states with highest infrastructure investment

**Dashboard Features & Analytics-**

**Core Dashboards**

1. **State Overview Dashboard**
   * Budget summary cards
   * Revenue vs Expenditure trends
   * Key financial ratios
   * YoY growth comparisons
2. **Comparative Analysis Dashboard**
   * Multi-state budget comparison
   * Ranking by various metrics
   * Correlation analysis
3. **Sector Analysis Dashboard**
   * Department-wise allocation
   * Sector performance trends
   * Budget utilization rates
   * Dynamic Filtering(state,year,category)
   * Priority sector analysis
4. **Financial Health Dashboard**
   * Deficit trends
   * Debt sustainability metrics
   * Revenue efficiency indicators
   * Fiscal performance scores

**Tech Stack:**

* **ETL Libraries**:
  + **Pandas**: Data cleaning and transformation for small datasets.
  + **SQLAlchemy**: Database interactions.
  + **openpyxl**: Excel file parsing.
* **Database**: **PostgreSQL** (scalable, supports partitioning, free).
  + **Why**: Handles large datasets (terabytes with indexing), robust SQL querying.
* **Dashboard Framework**: **Streamlit** (Python-based, easy-to-build web apps with file upload and Plotly integration).
* **Visualization**: **Plotly** (interactive charts: line for trends, bar for comparisons, pie for compositions).
* **RAG (Optional)**:
  + **LangChain**: Orchestrates AI queries.
  + **Ollama**: Runs local LLMs (e.g., Llama2) for free.
  + **FAISS**: Vector store for data retrieval.
* **Orchestration (Optional)**: **Apache Airflow** (for scheduled ETL runs, if needed).
* **Deployment**: **Docker** (containerize app + DB for portability),render/Heroku/streamlit cloud
* **Version Control**: **Git** (track code changes).
* **Monitoring/Logging**: **logging** for ETL logs; **Prometheus/Grafana** (optional for metrics).

**ETL Pipeline Details**

**1 Extract**

* **Parse multi-sheet Excel/CSV.**
* **Handle large files: Chunked reading (e.g., pd.read\_excel(chunksize=10000)).**
* **Enhancements: Metadata extraction (e.g., upload timestamp, file hash for idempotency).**

**2 Transform**

* **Cleaning: Handle NaNs (impute with median for numerics), standardize dates (fiscal year to TIMESTAMP).**
* **Computations: Time series ops (e.g., rolling averages, seasonality decomposition using statsmodels).**
* **Forecasting: Optional ARIMA or Prophet (open-source) for future predictions.**
* **Comparisons: SQL joins or Pandas merges.**