WRITTEN BY: - PARAG JYOTI NATH

PARK HOSPITALITY & VENTURES  JAIPUR, RAJASTHAN, INDIA

project:

centralized Database

[V-1.0]



**Documentation for Centralized Database Project**

The centralization of data originating from various systems—such as POS platforms (PetPooja, GoFrugal), accounting software (Tally), food delivery aggregators (Zomato, Swiggy, Fudr) and spreadsheet files (Excel)—necessitates a combination of technical and organizational competencies. The following documentation outlines the essential knowledge required and provides a breakdown of the technology stack and data flow involved in this project.

**🔧 Required Skills and Knowledge**

**1. Databases and Data Modeling**

* **SQL** is expected to be utilized for querying and managing relational databases (e.g., MySQL, PostgreSQL).
* Proficiency in **relational database design** is required to define entities and relationships effectively.
* **Normalization techniques** are to be applied in order to avoid redundancy and maintain data integrity.

**2. ETL (Extract, Transform, Load) Processes**

* Scripting languages such as **Python** or **Node.js** should be used to create routines that extract data from APIs or files and insert it into the centralized database.
* Optional use of ETL tools (e.g., **Apache NiFi**, **Talend**, **Airbyte**) may be considered depending on the scale of the integration.

**3. API and System Integrations**

* Familiarity with **REST API principles** is required for platforms like Zomato and Swiggy.
* API testing tools (e.g., **Postman**) should be used to validate endpoint responses.
* Knowledge of **OAuth and API key authentication mechanisms** is essential for secure access.

**4. Working with Files and Spreadsheets**

* Data parsing from **Excel or CSV files** should be conducted using libraries such as **Pandas**, **OpenPyXL**, or similar tools.

**5. Basic Accounting Knowledge**

* Understanding of accounting concepts (e.g., invoices, ledgers, and payments) is necessary for successful integration with Tally.

**6. Cloud Infrastructure (Optional but Recommended)**

* Hosting of databases and ETL pipelines is advised to be conducted on platforms such as **AWS**, **Google Cloud**, or **Microsoft Azure**.
* Alternatively, simpler backend solutions like **Supabase** or **Firebase** may be considered.

**7. Data Cleaning and Standardization**

* Custom scripts must be developed to **clean, transform, and standardize** diverse formats of raw data from different platforms.

**Optional but Useful Skills**

* Proficiency in **data visualization tools** such as **Power BI**, **Metabase**, or **Google Data Studio** is advantageous.
* Familiarity with **Git** for version control is recommended to manage the codebase efficiently.
* Experience with **Docker** is helpful for deployment and portability.

**🧰 Recommended Technology Stack**

**📊 1. Database (Core Data Layer)**

The centralized database is to be used for storing structured data originating from all integrated systems.

| **Tool** | **Use Case** | **Notes** |
| --- | --- | --- |
| **PostgreSQL** | Free, open-source relational database | Ideal for business and transactional data |
| MySQL | Widely adopted alternative | Strong community support |
| SQLite | Lightweight, file-based database | Suitable for prototyping or small-scale use |
| **Google Sheets / Excel** | Visualization or staging layer | Useful during early stages for output |

🟡 *It is recommended that beginners utilize* ***PostgreSQL****, possibly through managed services like* ***Supabase****, which offers a user-friendly interface and hosted environment.*

**🔄 2. ETL: Extract, Transform, Load**

The ETL process is responsible for collecting data from external sources, cleaning and transforming it, and subsequently loading it into the centralized database.

| **Tool / Language** | **Purpose** | **Justification** |
| --- | --- | --- |
| **Python** | Scripting for data ingestion | Extensive libraries: pandas, requests, openpyxl |
| Apache Airflow | Pipeline orchestration | Suitable for large-scale automation |
| Talend / N8N / Airbyte | Low-code or no-code ETL solutions | Good for teams with limited coding resources |
| Custom Scripts | Scheduled via cron or similar utilities | Lightweight and flexible |

🟡 *Python scripts are recommended to begin the implementation. For example:*

* PetPooja data → CSV → transformed using pandas → loaded into PostgreSQL
* Zomato data → API request via requests.get() → parsed → inserted into the database

**🧩 3. API and Platform Integration**

Data ingestion from the following platforms is to be carried out either through APIs or exports:

| **Platform** | **Integration Type** | **Comments** |
| --- | --- | --- |
| **Zomato** | Limited API | Use dashboard exports if API access is restricted |
| **Swiggy** | No public API | Rely on CSV or Excel exports |
| **PetPooja** | API + CSV exports | Provides reliable POS data |
| **GoFrugal** | API access available | May require vendor/partner access |
| **Tally** | XML / ODBC / Excel | Coordination with accountants may be needed |
| **Fudr** | Depends on subscription | Export options to be explored via support |
| **Excel** | Manual/automated upload | Can be parsed using openpyxl or pandas |

🟡 *Where APIs are unavailable, data exports in CSV/Excel format should be used, and processed via custom Python scripts.*

**📈 4. Dashboard and Reporting Tools**

These tools serve the purpose of data visualization and performance reporting:

| **Tool** | **Use Case** | **Beginner Friendly** |
| --- | --- | --- |
| **Metabase** | Open-source, connects to DB | ✅ |
| **Google Data Studio** | Visualization using Sheets or DBs | ✅ |
| Power BI / Tableau | Enterprise-grade dashboards | Moderate learning curve |
| Superset | Open-source alternative | Slightly technical |

🟡 *Begin with* ***Metabase*** *or* ***Google Data Studio*** *to generate meaningful insights quickly.*

**🧪 5. Optional Tools**

* **Supabase** – Hosted PostgreSQL with REST API and easy UI
* **DBeaver** – Cross-platform SQL client for managing databases
* **Visual Studio Code** – Preferred editor for scripting
* **GitHub** – Version control and project collaboration

**🛠️ Example Data Flow**

PetPooja CSV (daily export)

→ Python script reads data using pandas

→ Data is cleaned and standardized

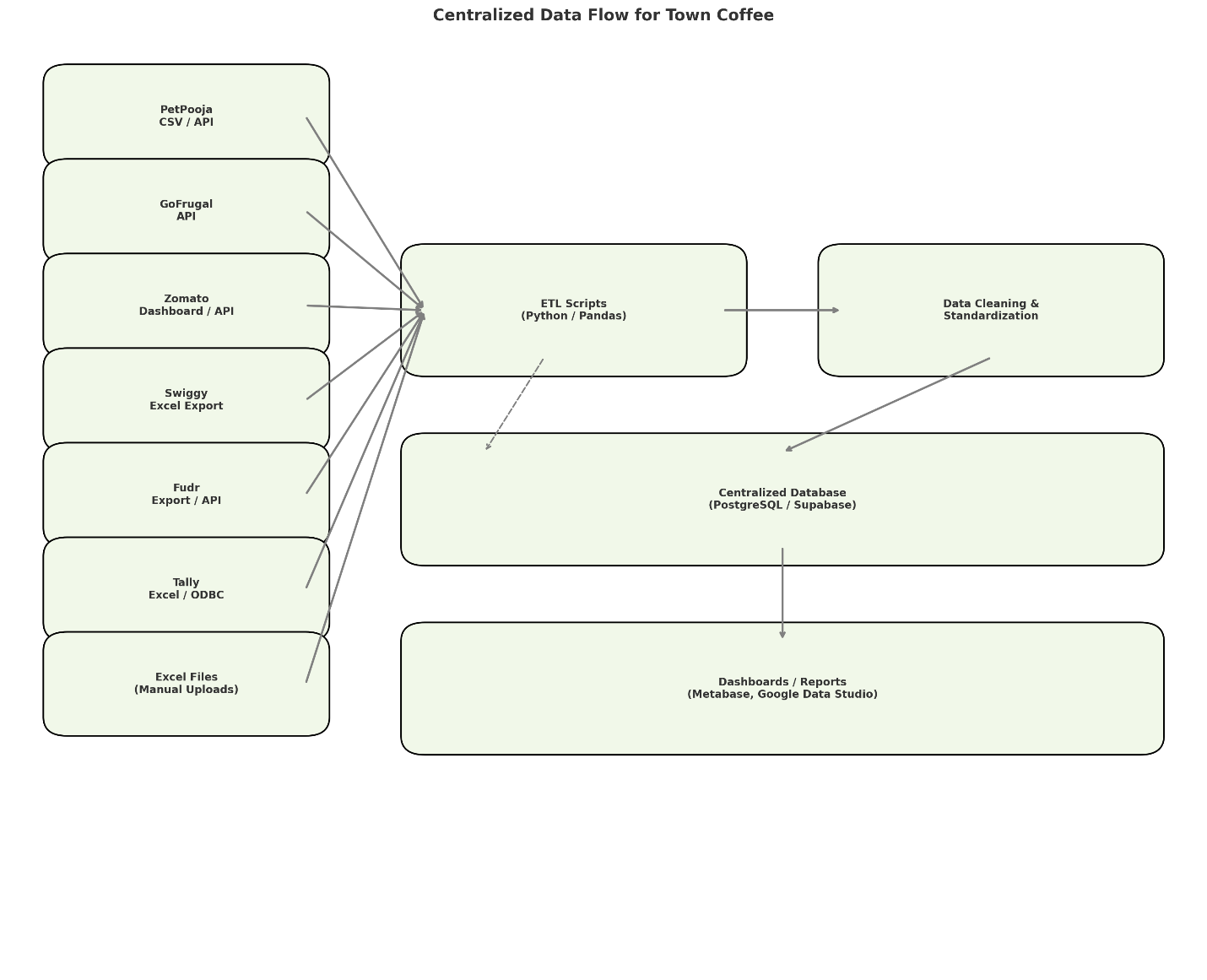
→ Data is inserted into PostgreSQL using psycopg2

→ Sales trends per outlet are visualized using Metabase

**✅ Next Step**

Should a sample Python script be provided for loading PetPooja or Excel data into PostgreSQL? Alternatively, the next phase could involve defining the database schema and mapping data from each platform into structured tables.

Here’s the **visual flowchart** of the centralized data pipeline for Town Coffee. It illustrates how data flows from multiple sources into a centralized database and finally into visualization tools.



Here is the updated visual data flow diagram with Fudr now included as one of the data sources. The arrows represent how data moves through the system:

1. Aggregators & Systems:  
   PetPooja, GoFrugal, Zomato, Swiggy, Fudr, Tally, and Excel files.
2. ETL Scripts:  
   Written in Python, used to fetch, transform, and prepare the data.
3. Cleaning & Standardization:  
   Optional layer to ensure data consistency across all sources.
4. Centralized Database:  
   PostgreSQL or Supabase is used as the main storage for unified data.
5. Dashboards/Reports:  
   Tools like Metabase or Google Data Studio are used to visualize insights.