**Project Overview**

The MCA regularly publishes company master data as state-wise CSV files on [data.gov.in](https://data.gov.in/), where I have selected 5 states- Maharashtra, Rajasthan , Goa, Tamil Nadu and Telangana. These files contain information such as Corporate Identification Number (CIN), company name, registration details, capital structure, status, and address. However, leveraging this data effectively is hampered by several factors:

* Data Fragmentation: Information is split across multiple files for different Registrar of Companies (RoC) jurisdictions, requiring manual effort to create a unified view.
* Dynamic Nature: The dataset is a snapshot in time. Tracking changes—such as new incorporations, companies being struck off, or changes in capital—across subsequent releases is a manual, time-consuming, and error-prone process.
* Lack of Context: The raw data lacks enriched information (e.g., director details, sector classifications, website links) that is often essential for deeper analysis.
* Accessibility for Non-Technical Users: Data analysts and business users need an intuitive way to query and visualize this data without writing complex database queries.

The MCA Insights Engine was conceived to solve these problems, providing a scalable, automated, and intelligent platform for corporate data monitoring.

**Project Objectives**

The primary objective was to design and implement a system that performs the following core functions:

1. Data Integration & Normalization: Automatically merge multiple state-wise CSV files into a single, clean, and standardized master dataset.
2. Change Detection & Audit Logging: Systematically identify and log changes between data snapshots, categorizing them into 'New Incorporations', 'Removals', and 'Updates'.
3. Public Data Enrichment: Augment newly identified companies with supplementary data scraped from public online sources like ZaubaCorp.
4. Interactive Data Access & Visualization: Develop a user-friendly web dashboard for searching, filtering, and visualizing data through charts and metrics.
5. AI-Powered Intelligence: Implement a conversational AI chatbot that allows users to ask questions in natural language and receive synthesized answers based on the live database.

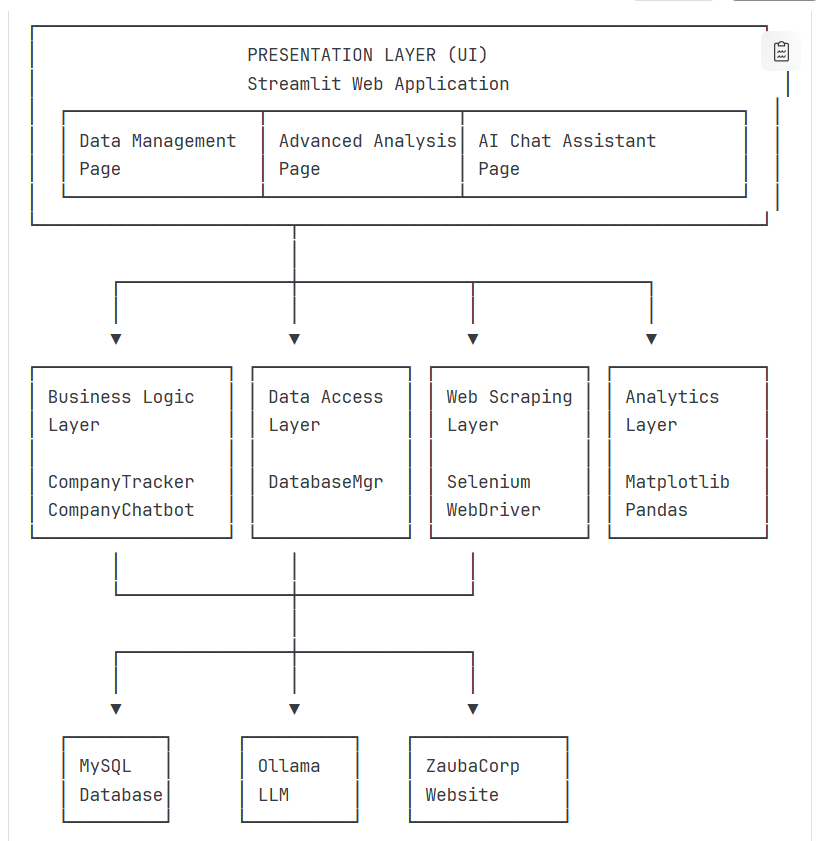
**Technology**

The platform is built on a Python foundation for data processing, with MySQL providing persistent data storage structured to support audit requirements. Selenium automates web-based enrichment through browser automation that includes sophisticated anti-detection measures. Streamlit provides the web dashboard interface, while Ollama handles conversational AI requests locally without requiring external API dependencies.

**System Design & Architecture**

The system employs a multi-tiered architecture to ensure modularity, scalability, and clarity of separation of concerns.

**Architecture Diagram**



**1. Data Layer (MySQL)**  
The relational database is designed with three core tables:

* companies: Serves as the master table, holding the most current record for every company. The sync\_flag column indicates if a record is 'NEW' or 'UNCHANGED'.
* change\_log: An immutable audit trail. It stores a complete copy of a company's data at the moment a change was detected, along with the change\_type ('NEW', 'REMOVED') and a timestamp. This design provides a full historical record.
* companies\_archive: Stores the complete records of companies that have been removed from the master table, including the reason and timestamp of archiving.

**2. Processing Layer**   
This layer contains the core business logic, implemented in several Python classes:

* CompanyTracker: The orchestrator class. It manages the entire workflow: file ingestion, data cleaning, change detection, and interaction with the database.
* DatabaseManager: Abstracts all database operations. It handles connection pooling, schema initialization, and complex transactions (e.g., bulk inserts, archiving, and change logging), ensuring data integrity.
* Data Cleaning Pipeline: A robust process that:
  + Standardizes column names.
  + Validates and cleans critical fields like CIN and dates.
  + Generates a unique row\_hash for each record using an MD5 hash of key fields, which is pivotal for efficient change detection.
  + Removes duplicate records based on CIN.

**3. Enrichment Layer (Selenium Web Scraper)**

* zauba\_scraper.py contains functions to automate browser interaction.
* For each new company, it programmatically navigates to the relevant ZaubaCorp search page, extracts available details (name, address, profile URL), and compiles them into an enriched Excel report.

**4. Presentation & AI Layer (Streamlit & Ollama)**

* Streamlit Dashboard: Provides a cohesive web interface with three main pages:
  1. Data Management: For file upload, processing, and viewing master/change data.
  2. Advanced Analysis: For interactive data exploration through visualizations.
  3. AI Chat Assistant: For natural language interaction with the data.
* CompanyChatbot Class: Integrates with a local LLM (Llama 3.2 via Ollama). It provides the LLM with database context (schema, recent stats) and user questions, enabling it to generate accurate, context-aware responses and even suggest SQL queries.

**Installation and Setup**

**Prerequisites**

The system requires Python 3.8+, MySQL 8.0+, Chrome browser for automation, and 4GB minimum RAM. Internet connectivity is essential for external data enrichment.

**Setup Process**

Installation involves creating a Python virtual environment, installing dependencies from requirements.txt, configuring MySQL database connection parameters through environment variables or config files, and launching the Streamlit dashboard. The system auto-creates required database tables on first connection or can be manually initialized through schema scripts.

ChromeDriver for browser automation is automatically downloaded and managed through the webdriver-manager dependency, eliminating manual driver management complexity.

**Data Ingestion Workflow**

The data ingestion process follows a structured five-phase workflow designed to handle diverse data quality issues.

1. **Phase 1: File Upload and Validation** - Users select CSV files through the dashboard and review previews before processing, enabling manual verification of file selection.
2. **Phase 2: Multi-Strategy Reading** - The system applies progressive reading strategies to handle encoding variations. It attempts UTF-8 first, then sequences through Python engine parsing, bad-line skipping, Latin-1 encoding, and manual CSV parsing with proper quote handling. Success at any stage halts fallback attempts.
3. **Phase 3: Data Normalization** - Column names are standardized, data types enforced (with CIN validation), capital amounts cleaned of formatting characters, dates converted to ISO format, and text fields trimmed. The system generates quality metrics documenting cleaning operations.
4. **Phase 4: File-Level Deduplication** - MD5 fingerprints are generated for each record using CIN, company name, paid-up capital, company status, and authorized capital. Duplicates are identified and removed, keeping only the first occurrence.
5. **Phase 5: Multi-File Consolidation** - All deduplicated files are merged into a master dataset, consolidation-level deduplication removes cross-file duplicates, and the consolidated dataset is validated and saved as staging data.

After ingestion, the system generates comprehensive quality metrics including record counts before and after cleaning, duplicate removal statistics, null value analysis, and overall quality scores.

**Change Detection Logic**

1. **Detection Algorithm**

The system compares the new consolidated dataset against existing database records using CIN as the matching key. Set operations identify three categories:

1. **New Incorporations** - CINs in new data but absent from database. For each, the system creates a change log entry with complete company state, assigns sync flag "NEW", and prepares for database insertion. This creates an auditable record of when companies entered the system.
2. **Deregistrations** - CINs in database but absent from new data. For each, the system archives the complete record to the archive table with removal reason, creates a change log entry, and deletes from the active table. This preserves historical data while maintaining a current working dataset.
3. **Unchanged Records** - CINs in both datasets marked with sync flag "UNCHANGED". The architecture supports future enhancement to detect field-level modifications.
4. **Audit Trail Design**

All change log entries preserve complete company state at each change point rather than only storing differences. This enables historical reconstruction and supports regulatory compliance requirements. Timestamps and change type indexes enable efficient querying of specific change categories within date ranges.

If duplicate CINs appear in the new dataset, only the first occurrence is processed, with duplicates logged as warnings for investigation.

**Results & Outputs**

The implemented system successfully delivers all required functionalities:

* Unified Master Data: A single source of truth for company data is maintained within the MySQL companies table.
* Comprehensive Change History: The change\_log table provides a permanent, queryable audit trail of all corporate lifecycle events, fulfilling the need for auditable change tracking.
* Enriched Data Samples: The system successfully demonstrates the automated scraping of ZaubaCorp, producing structured Excel files (added\_enriched.xlsx) with enhanced details for newly incorporated companies.
* Interactive Visual Analytics: The dashboard provides over eight different chart types, enabling users to visually explore trends, distributions, and correlations within the data without any technical expertise.
* Conversational Data Access: The AI chatbot effectively answers a wide range of natural language questions, such as "List new companies in Maharashtra" or "What is the status distribution?", making data exploration accessible to all users.

**Conclusion**

The MCA Insights Engine project successfully transitions the concept of static MCA data monitoring into a dynamic, intelligent, and automated process. By integrating a robust ETL pipeline, a secure database, a responsive web dashboard, and a powerful AI layer, the system delivers a comprehensive solution that addresses the core challenges outlined in the problem statement. It not only saves significant manual effort but also unlocks deeper, more accessible insights from the MCA dataset. This platform serves as a strong foundation for future enhancements, such as integrating additional data sources (MCA API Setu, GSTN), predicting company status changes using machine learning, and scaling to handle all-India data.