Project Documentation: AgenticAi – Real-Time Governance Data Analyzer

1. Abstract

Governments and organizations release large amounts of public data, but much of it is unstructured, inconsistent, and underutilized. Traditional data cleaning and analysis pipelines are time-consuming, reducing the effectiveness of data-driven decision-making.

AgenticAi solves this by automating the **end-to-end lifecycle of dataset processing**: ingestion, cleaning, standardization, and instant insights. It provides a command-line and backend-driven solution that turns raw datasets into **actionable, queryable intelligence**.

2. Introduction

2.1 Problem Statement

- Public datasets are often inconsistent (typos, missing values, mismatched columns).
- Manual cleaning takes **hours to days**, delaying insights.
- Policymakers, researchers, and developers need **faster, reproducible pipelines** to use open data effectively.

2.2 Objectives

- Build a system that can work with **any dataset, any sector, any format**.
- Automate **cleaning, transformation, and validation**.
- Provide **instant exploratory analysis** (tables, summaries, ASCII charts).
- Allow **querying datasets in plain English** (planned AI extension).

3. System Architecture

3.1 Components

- **Backend (Node.js + Express)**
    - API endpoints for dataset upload and cleaning
    - Data pipeline orchestration
- **Data Processing (Python)**
    - Cleaning, transformation, and standardization using **Pandas**
    - Insights and visualization (Matplotlib, Tabulate)

- **Frontend (React)**
    - User-friendly dashboard for dataset uploads and queries
- **AI Integration (LangChain)**
    - Agentic AI layer for **natural language querying** of datasets

3.2 Workflow

1. User uploads dataset (CSV).
2. System cleans and standardizes data automatically.
3. Insights are generated (row counts, top categories, trends).
4. Data is stored and can be queried through CLI or API.
5. (Future) Users ask questions in plain English → AI returns insights.

4. Features

- Upload any CSV dataset
- Automated data cleaning (typos, missing values, column normalization)
- Standardized output CSV
- Exploratory insights (summary tables, ASCII visualizations)
- Step-by-step logs for transparency
- Natural language Q&A over datasets
- Web dashboard for non-technical users

5. Tech Stack

- **Languages**: Python, JavaScript
- **Libraries**: Pandas, Matplotlib, Tabulate, Express.js
- **Database**: MongoDB
- **AI**: LangChain, LLMs (Groq/OpenAI)
- **Hosting**: Node.js server, Python services

6. Dataset (Example: Telangana MSME Data)

- Columns: district, mandal, industry_category, type_of_industry, unit_name, investment, employment, status, export, type_of_connection
- Issues: typos (indsutry_category), missing values, inconsistent formatting

- Cleaned Output: telangana_msme_cleaned.csv

7. Installation & Usage

```
# Clone repository
git clone https://github.com/VarshiniNeralla/AgenticAi.git
cd AgenticAi

# Install backend dependencies
cd backend
npm install

# Start backend
node server.js

# Run data cleaning in Python
cd data-processing
python clean_dataset.py --file your_dataset.csv
```

8. Results

- Raw MSME dataset cleaned into a standardized format
- Errors fixed automatically (typos, missing values)
- CLI-based insights generated instantly
- Foundation for real-time governance dashboard

9. Future Scope

- Web dashboard for non-technical users
- AI-powered question answering on datasets
- Integration with multiple public APIs
- Visualization dashboards for policy decisions

10. Conclusion

AgenticAi demonstrates how **automation + AI** can unlock the real value of public datasets. By reducing data preparation time from hours to minutes, it empowers policymakers, researchers, and developers to **make faster, evidence-based decisions**.