

Data-Talk-AI: “Chat With Your Data Files VIZ CSV, XLSX Effortlessly”

Harsh Katariya

Dept. Information Technology

Shree L R Tiwari College of Engineering

Mumbai, India

harsh.katariya@slrtce.in

Sanket Mithbavkar

Dept. Information Technology

Shree L R Tiwari College of Engineering

Mumbai, India

sanket.s.mithbavkar@slrtce.in

Omkar Kadam

Dept. Information Technology

Shree L R Tiwari College of Engineering

Mumbai, India

omkar.a.kadam@slrtce.in

Yogesh Arunthuthiyar

Dept. Information Technology

Shree L R Tiwari College of Engineering

Mumbai, India

yogesh.t.arunthuthiyar@slrtce.in

Amruta Sankhe

Dept. Information Technology

Shree L R Tiwari College of Engineering

Mumbai, India

amruta.sankhe@slrtce.in

Abstract— The task of collecting and examining raw data is often difficult for both people and businesses who do not have any technical knowledge on how to code or have the finances to purchase expensive data analysis tools. The current solutions require users to know how to use programming languages (i.e., Python and R) or require the use of cloud services where many users have concerns about privacy, cost, and access to the internet. Growing differences between the amount of data and the ability to use data led to the creation of Data-Talk-AI, which is a user-friendly, web-based application that will allow customers to interact with their datasets using everyday conversation in plain English while not writing any code.

The Data-Talk-AI product uses a locally cached Qwen 2.5 (large language model)-built on Ollama- with a Streamlit UI, to provide an all-inclusive offline, privacy-preserving data cleaning and analysis solution. There are 11 steps Rule-Based automated data cleaning (missing rows/duplicate rows/outliers/sentinels/missing data type/mixed type/inconsistent format) with one button click. Users clean their data before requesting interactive visualizations via natural language, and the product automatically determines what type of chart to use/sets up axis labels & legend/configures the context based on the generated chart + provides the user with AI-generated context-based insights from the chart

created. Data-Talk-AI uses MySQL-based authentication and bcrypt hashed password storage to operate fully on local infrastructure (no reliance on any cloud services) and provides non-tech professionals with professional-grade data management.

Keywords— *Natural Language Processing, Data Cleaning Automation, Large Language Model, Local AI Inference, NLP-Driven Visualization, Data Quality Management*

I. INTRODUCTION

In recent years, the ability to transform raw data into meaningful insights has become increasingly vital across domains such as education, research, and business. Data visualization, in particular, serves as a powerful means to uncover hidden patterns and communicate complex findings in an intuitive way. However, creating accurate and useful visualizations often requires significant technical expertise, including familiarity with programming languages, query formulation, and specialized visualization libraries. This dependency creates barriers for non-technical users, limiting their ability to fully leverage the potential of data [1]. Natural language interfaces have been studied as a promising way to reduce this barrier, allowing users to specify analytical or visualization tasks in everyday language. Prior research highlights challenges such as accurately interpreting user intent, handling heterogeneous datasets, and ensuring that

generated outputs remain both correct and readable. While rule-based and early machine learning approaches offered partial solutions, they often struggled with generalization and scalability. The advent of large language models (LLMs) has opened new opportunities by enabling more robust interpretation of natural language queries, code generation, and automated visualization. Recent systems such as Chat2Vis, LIDA, and ChartLlama demonstrate that LLMs can generate charts directly from natural language prompts combined with tabular data, yet they still face issues related to code execution, incorrect data transformations, and limited evaluation practices [2].

The application of LLMs to geospatial science involves leveraging these powerful language models to analyze, understand, and generate information related to geographic and spatial data[9]. As a support tool for data exploration, a CA could enable citizens to participate in this research step without producing additional work or costs for initiators, such as personal training or mentoring. Also, it would be a more scalable and constantly available solution, quickly providing citizen scientists with information and assistance to enable their participation and learning[5].

II. EASE OF USE

When designing Data-Talk-AI one of the major considerations was making it accessible to anyone regardless of their prior experience with programming. Instead of forcing users to learn the syntax of a programming language or navigate complex setup configurations for usage, Data-Talk-AI simply accepts everyday language requests such as “show me the missing values in the Age column” or “create a bar chart of Sales by region” to accomplish the desired task without requiring any additional actions from the user. Data-Talk-AI contains a simple, browser-based interface requiring little to no installation of anything other than a few necessary Python packages and an Ollama running locally, and has a sidebar-driven set of common actions to easily upload files, trigger the auto-cleaning function and reset data all at the click of a

button. Users also receive immediate feedback when using Data-Talk-AI because the AI uses streamed word-by-word responses, logs all of the cleaning steps in real-time, and generates and displays any requested visualizations immediately upon completing each task giving the user confidence that they are receiving timely and appropriate assistance from the system vs quietly waiting for the system to return its results to them.

III. TECHNOLOGIES USED

Sr. No.	Technology Used	Category	Purpose
1	Python 3.11	Core Language	Backbone of the entire application; handles all data processing, LLM integration, authentication, and business logic
2	Streamlit	Web Framework	Builds the complete browser-based frontend in pure Python — file upload, chat, visualization, sidebar, and export tabs
3	Pandas	Data Processing	Reads CSV/Excel files, powers the 11-step auto-clean pipeline including null imputation, duplicate removal, and type coercion
4	NumPy	Numerical Computing	Supports outlier detection via IQR, trendline fitting in scatter charts, and array-based statistical operations
5	Plotly	Visualization	Renders 10+ interactive chart types (bar, line, scatter, heatmap, pie, box, etc.) with zoom, hover, and export support
6	Ollama	LLM Runtime	Runs Qwen2.5 locally on the user's machine;

			exposes streaming API for all AI inference without any cloud dependency
7	Qwen2.5	Large Language Model	Handles NLP chat, parses plain English into JSON chart configs, and generates contextual insights from rendered charts
8	MySQL	Database	Stores registered user credentials locally; handles all login and signup queries via a dedicated users table
9	bcrypt	Security	Hashes user passwords using adaptive cryptographic algorithm before storing in MySQL; prevents plaintext credential exposure
10	mysql-connector-python	DB Driver	Official Python-MySQL bridge; manages connections, executes SQL queries, and handles authentication errors gracefully
11	Custom CSS & Inter Font	UI Styling	Hand-crafted design system injected via Streamlit; delivers white/blue professional theme with chat bubbles, badges, and skeleton loaders
12	Plotly JS (Frontend)	UI Interactivity	Powers browser-side chart interactions — hover tooltips, zoom, legend toggle, and image download without any JavaScript development

Fig.3.1 Tech-Stack of Data-Talk-AI

The Above Table 3.1 shows that Data-Talk-AI project is based upon a deliberately chosen technology stack, comprising Python 3.11 as the host

programming language, Streamlit as the full-stack web framework, and Pandas and NumPy as the two main engines used in an eleven-step rule based automated cleaning pipeline for data processing. Interactive visualizations will be implemented using Plotly and all AI inference will run locally using Ollama's Qwen2.5 large language model, providing 100% data privacy with no dependency on any cloud services. User authentication is implemented through a MySQL database with bcrypt hashed passwords using the mysql-connector-python driver, and the Data-Talk-AI interface will use a custom CSS design system based on the Inter typeface, resulting in a lightweight, offline-first and privacy-preserving platform that will operate completely on locally hosted resources without any configuration overhead.

IV. LITERATURE SURVEY

Chen et al. introduce VisEval, a comprehensive benchmark addressing the critical gap in systematic evaluation of large language models (LLMs) for Natural Language to Visualization (NL2VIS) tasks. Despite advancements in LLM-based methods like Chat2vis, LIDA, and specialized models such as ChartLlama and ChartGPT, current state-of-the-art LLMs including GPT-4, GPT-3.5, Gemini-Pro, and CodeLlama still generate visualizations plagued by code execution failures, incorrect data transformations, illegal sorting, and readability issues. Existing NL2VIS evaluation practices suffer from limitations in dataset quality, narrow domain coverage, incorrect labels, and inadequate evaluation metrics that fail to address multiple quality dimensions beyond data correctness. VisEval contributes a high-quality dataset of 2,524 representative queries across 146 databases with accurately labeled ground truths, constructed through rigorous filtering combining LLM intelligence with visualization expert experiences. The benchmark implements a novel automated evaluation framework assessing three critical dimensions: validity (code execution capability), legality (compliance with query requirements through systematic checks of chart type, data transformation, visual mapping, and sorting), and

readability (layout issues and comprehension factors leveraging GPT-4V's multimodal capabilities, achieving 0.843 Spearman correlation with human ratings). Comprehensive evaluation reveals GPT-4 achieves the highest quality score of 2.89/5 with 75.27% pass rate using Matplotlib, while all models show declining performance on complex visualizations requiring three visual channels, with typical errors including invalid code, illegal data/visualization transformations, incorrect ordering, and low readability issues. The research demonstrates that table format representation significantly impacts performance (pass rates varying by 2.88%) and including unused tables degrades performance, while VisEval's modular design enables extension to JavaScript-based toolkits and Vega-Lite methods, positioning it as essential infrastructure for advancing automated data visualization generation[3].

Li et al. present PROMPT4VIS, a framework leveraging large language models (LLMs) with in-context learning (ICL) to transform natural language queries into data visualization commands for tabular data, addressing the critical limitation where existing methods like Seq2Vis, ncNet, and RGVisNet achieve over 90% accuracy in domain-specific scenarios but plummet below 30% in cross-domain settings. The framework comprises two core components: an example mining module that optimally selects demonstration examples by jointly considering similarity (Euclidean distance between sentence embeddings), influence (measured via contrastive learning-based model using BERT encoding with cross-entropy loss), and diversity (average distance between candidates) through a weighted objective function proven NP-hard via reduction from Maximum Coverage Problem, solved with a greedy approximation algorithm achieving $(1-1/e)$ approximation guarantee due to submodularity; and a schema filtering module that streamlines database schemas by prompting LLMs to select necessary tables, eliminating irrelevant information. Evaluation on NVBench dataset across five data partitions demonstrates PROMPT4VIS with GPT-3.5-Turbo

backbone significantly outperforms state-of-the-art RGVisNet by approximately 71.3% on test sets and 35.9% on development sets in overall accuracy, achieving 52.69% and 60.39% respectively, with weighted scores showing 40.94% versus RGVisNet's 14.14% across different query hardness levels (extra hard, hard, medium, easy). Ablation studies confirm strategic example selection outperforms random selection or similarity-only approaches, while schema filtering achieves 65-70% precision in target schema identification compared to RAT-SQL's 39-44% (table-level) and 22-25% (column-level), though providing complete schemas causes 4.3-6.6% performance degradation due to information overload. The framework demonstrates robustness to early-stage errors and manual parameter tuning, with performance enhancement possible through text-to-SQL integration via fine-tuning models like Qwen2.5-Coder-14B-Instruct, though simple chain-of-thought prompting proves insufficient. Error analysis reveals struggles with high-hardness examples requiring extensive data manipulation, particularly in axis label mapping and function application, with LLM hallucination occasionally producing incorrect WHERE clauses or illegal grammar. PROMPT4VIS marks the first ICL-based framework for text-to-vis tasks, demonstrating a paradigm shift from complex neural networks to adaptive approaches, with time complexity $O(N_{\text{train}} \log N_{\text{train}}) + O(k \cdot N_{\text{train}})$ and space complexity $O(N_{\text{train}}^2)$, though scalability limitations emerge beyond 100,000 training samples[2].

Zhang et al. provide a concise yet comprehensive survey of natural language interfaces (NLIs) for tabular data querying and visualization, highlighting the shift from manual SQL and plotting to intuitive text-based interactions powered by semantic parsing and large language models (LLMs). They trace dataset evolution from single-domain benchmarks (ATIS, GeoQuery) to cross-domain (WikiSQL, Spider), conversational (SParC, CoSQL), and multilingual/extensions (CSpider, DuSQL) constructs, alongside Text-to-Vis datasets like nvBench and ChartDialogs, underscoring the need for diverse,

realistic evaluation. Methodologies progressed from rule-based systems (PRECISE, NaLIR) to neural encoders—sequence-based (TypeSQL), graph-based (RAT-SQL)—and decoders ranging from monolithic skeletons to grammar- and execution-based models (SyntaxSQLNet, PICARD), before entering the foundation-model era with BERT- and T5-based approaches and LLM prompting strategies (C3, Prompt4Vis). Evaluation metrics span string-based matching, execution-based comparisons, and human studies to capture semantic fidelity, while system architectures evolved from rigid rule-based engines to parsing pipelines, multi-stage modules, and end-to-end frameworks balancing accuracy and adaptability. The authors identify gaps in Text-to-Vis research—particularly its lag behind Text-to-SQL—and advocate for leveraging LLM customization, advanced learning paradigms beyond supervised training, improved domain generalization, and richer, multi-dimensional evaluation frameworks to drive future innovation[4].

Wang et al. propose a GPT-based Text-to-SQL framework for spatial databases that overcomes the scarcity of labeled spatial datasets and the limitations of rule-based methods by integrating geographic and database-specific knowledge into prompt design. They introduce a bilingual English–Chinese dataset comprising 200 natural language questions paired with 296 SQL queries across four real-world spatial databases (Ada, Edu, Tourism, Traffic), with and without derived columns, to evaluate spatial and general queries under 0-, 1-, 3-, and 5-shot settings. Their prompt consists of six components—SDBMS knowledge, in-context examples, schema code, geographic descriptions, schema tips from execution feedback, and the target question—to guide GPT-4 (and DeepSeek) through a self-correcting loop that retries on errors until success or a repetition limit. Experiments demonstrate that their method (SSpa) achieves up to 87.5% execution accuracy on spatial SQL tasks, outperforming a state-of-the-art relational baseline (DAIL-SQL) by over 22% and validating the critical role of each prompt component via ablation studies. They also analyze error patterns, highlight

scalability and function-coverage limitations, and outline future improvements including automated function selection, richer schema feedback, and dataset expansion to further enhance spatial Text-to-SQL performance[1].

Stein et al. develop a conversational agent (CA) to support data exploration in citizen science, tackling barriers of data literacy and engagement through a design science research approach that yields five design principles and a prototype evaluated experimentally. They derive design principles ensuring structured guidance (menu-driven workflow), tiered knowledge (layered explanations), ad hoc Q&A (user questions answered or forwarded), integration of internal and external learning materials, and proactive warnings of common pitfalls, all grounded in user needs from expert workshops and literature on data literacy, citizen science, and agent design. The prototype, implemented in Rasa with a Telegram frontend, offers two workflows: a menu-based data analysis process for beginners and a Q&A flow for advanced users, leveraging messages, links, and images to provide dataset-independent support. In an online experiment ($n=30$, between-subjects), CA users outperformed controls by 14% in task performance ($p=0.048$) on Titanic dataset exploration tasks, though no significant differences appeared in perceived learning, empowerment, or motivation; qualitative feedback highlighted varying user paths, demand for dataset-specific Q&A, and suggestions for improved onboarding and interactivity. The study contributes five design principles for educational CAs in data exploration, demonstrates the CA's positive impact on actual performance, and outlines future work on enhancing hedonic quality, bridging perceived learning gaps, and deploying the CA in real-world citizen science contexts[5].

Ko et al. propose VL2NL, a three-stage large language model (LLM) framework that generates diverse and high-quality natural language (NL) datasets for data visualization by leveraging Vega-Lite specifications as input, significantly addressing limitations in previous crowdsourcing approaches and

synthetic collections. Their pipeline introduces guided discovery in LLM prompts to accurately synthesize chart semantics and incorporates score-based paraphrasing along four language axes—subjectivity, formality, clarity, expertise—to improve NL diversity and realism, demonstrated through the largest collection to date of 1,981 real-world, human-generated Vega-Lite charts spanning high complexity, numerous chart types, interactions, and composite views. VL2NL produces NL datasets for captioning (L1, L2), chart generation utterances (command, query, question), and question answering, with its output surpassing benchmark and gold-standard baselines on metrics of syntactic diversity, semantic coverage, and objective annotation accuracy (e.g., 89.4% for chart semantics, 76% for captions). Paraphrased NL datasets enhance lexical and structural variety, capture broader user behaviors, and boost downstream machine learning model performance in chart-type classification and other visualization tasks, as confirmed by finetuning experiments and expert studies on fully-automatic and mixed-initiative modes. The work concludes with a call for augmenting Vega-Lite specifications, extending conversational NL datasets, and leveraging external reasoning tools to further advance NLI for data visualization, positioning VL2NL as a scalable foundation for future research and practical deployment in visualization interfaces[6].

Achi Mordechai et al. introduce NoviCode, a novel Natural Language Programming (NLProg) benchmark and framework targeting the generation of executable, complex programs from intuitive, everyday natural language utterances by non-programmers, using API specifications as the bridge between high-level goals and low-level code. Unlike prior Text-to-Code approaches, NoviCode focuses on novice, non-technical language (devoid of programming jargon) and demands control-flow-rich outputs—sequences, conditions, loops—often only implicitly referenced in the input, making both translation and evaluation substantially more challenging. The benchmark is built from 1,200

crowd-sourced English utterances spanning nine domains (such as calendar, messaging, smart home, and weather), paired with 150 expert-written functional test suites and a domain-specific Python API, allowing assessment based on actual execution rather than code form. The authors augment the benchmark with a scalable, grammar-driven synthetic training set and propose a novel hierarchical modeling paradigm, where models first generate compact abstract syntax trees (cASTs) capturing the structural logic of target programs before deterministic conversion to code, in contrast to standard end-to-end sequence models. Experimental results across multiple large language models (GPT-4 Turbo, GPT-3.5 Turbo, CodeLlama, Mistral, DeepSeek Coder, T5, CodeT5, and CodeT5+) show that the hierarchical approach using cASTs consistently outperforms simple text-to-code baselines in both in-context and fine-tuned scenarios—GPT-4 Turbo achieves 39% pass@1 with cAST versus 33.8% without, and CodeT5+ reaches 11.7%—although loops and implicit control flows remain particularly difficult for all models. Extensive error analysis reveals the main failure modes as hidden (implicit) control flow, logical and semantic errors more than syntax, and the persistent difficulty in mapping colloquial requests to proper API actions. NoviCode thus highlights both the progress and the open challenges of natural language programming for laypeople, offering a scalable task, robust evaluation, and a path forward for advancing human-centric code generation systems[7].

Kavaz et al. present a comprehensive scoping review of chatbot-based natural language interfaces (V-NLIs) for data visualisation, analysing 20 recent systems to map out advances, challenges, and opportunities across the full data visualisation pipeline—data, visual, and interaction spaces. The review characterises V-NLIs by their ability to handle data transformation (most supporting multidimensional tabular data, some complex types like networks and hierarchies), visual mapping (where basic chart types predominate, with limited support for advanced layouts and user-defined or intelligent

mapping), and a wide range of interaction methods (select, filter, encode, and reconfigure, mostly through multimodal input and output, but with few systems enabling high-level or follow-up queries). Chatbot-based interfaces, compared to form-based NLIs, enrich the user experience by providing conversational guidance, textual and visual feedback, and support for recommendations and query auto-completion, although guidance is mostly restricted to low-level operations and interface discoverability remains a challenge. The survey identifies critical limitations in existing V-NLIs, such as poor support for complex data and queries, lack of automation in visual mapping, and limited integration with advanced technologies like AR/VR; and recommends future research directions in expanding anthropomorphic and intelligent agent capabilities (AINT framework), improving feedback and narrative storytelling, supporting multimodal and collaborative analytics, and leveraging large language models for sophisticated interaction and explanation. This work offers a taxonomy, benchmarks state-of-the-art approaches, and sets a clear agenda for moving V-NLIs toward more intelligent, adaptive, and collaborative visual analysis tools[8].

V. RELATED WORK

The development of natural language interfaces for data analysis has seen significant advancements, particularly with the integration of large language models (LLMs) and interactive frameworks. Several systems and surveys have laid the groundwork for platforms like Data-Talk-AI.

Chat2VIS: Generating Data Visualizations via Natural Language

Chat2VIS is a cutting-edge technology that gives you the power to create data visualizations without coding. By leveraging the capabilities of large language models (LLMs), such as GPT-3, Codex, and ChatGPT, Chat2VIS allows you to enter a request in plain English and receive code that will create your desired visualization. Because of the effectiveness of

the prompt engineering used, Chat2VIS can produce visualizations even when the user does not provide a complete or correct response to their question – i.e., they have mis-specified or under-specified input. These factors demonstrate that LLMs are capable of performing end-to-end visualization tasks from start to finish and provide a low-cost and scalable alternative to traditional visualizing methods that depend on human-generated rules and grammars. In addition, Chat2VIS addresses privacy issues by architecting all of its prompts so that no user information can be discovered by Chat2VIS through its use of LLMs[11].

Chat2Data: An Interactive Data Analysis System with RAG, Vector Databases, and LLMs

Chat2Data has developed a three-layer architecture for interactive data analysis, which consists of three layers: retrieval-augmented generation (RAG), vector database, and LLMs. This system will help provide a higher degree of accuracy and efficiency for natural language queries by embedding domain knowledge into each layer of the retrieval-augmented generation (RAG) process and minimizing the number of calls to LLMs within the system. By embedding RAG within the architecture, it addresses issues such as hallucination (RAG will ensure that generated responses are truthful) and will result in creating more reliable systems for performing complex analytical tasks[12].

Towards Natural Language Interfaces for Data Visualization: A Survey

This research paper presents a detailed analysis of a variety of different natural language interfaces (NLIs) designed to allow users to interact with and visualize data. The systems mentioned in the paper represent different stages of classical information visualisation pipelines including query interpretation, transforming data, mapping data to visual form, and presenting a visual representation. Various design challenges that affect the user experience including interpreting the user's true intentions and providing a visual representation that can be created from an ever-

changing set of heterogeneous datasets were addressed. Promising avenues for future research within the NLI community have been proposed. Overall, this research provides a comprehensive understanding of the design and evaluation metrics for NLI system[13].

Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey

This survey looks at how natural language interfaces have progressed for using conversational language to query and visualize tabular datasets. It summarizes recent advances in semantic parsing, explains how large language models play into that advancement, discusses the creation of both datasets and evaluation metrics, and identifies both the gaps and strengths of the currently available systems while also suggesting opportunities for improvement. It provides a roadmap for both researchers and practitioners who are working towards building more user-friendly and effective natural language interface systems for interacting with data[4].

VI. PROPOSED SOLUTION

This paper proposes Data-Talk-AI, an NLP-driven data cleaning and analysis platform that enables users to interact with their datasets through plain conversational English without requiring any programming expertise or internet connectivity. The system integrates a Qwen2.5 large language model served via the Ollama runtime with a Streamlit-based web interface, supported by an eleven-step automated cleaning pipeline built on Pandas and NumPy that autonomously resolves missing values, duplicates, outliers, mixed data types, sentinel strings, and inconsistent formatting in a single operation.

Beyond automated cleaning, the platform incorporates a natural language visualization engine that translates plain English chart requests into structured Plotly visualizations with AI-generated contextual insights, alongside a MySQL-backed authentication system with bcrypt password hashing

for secure multi-user access. All data processing, model inference, and storage operations are executed entirely on the user's local machine, ensuring complete data privacy with zero dependency on external cloud services or paid APIs, making Data-Talk-AI a practical and privacy-preserving alternative to existing cloud-dependent data management tools.

VII. METHODOLOGY

Data-Talk-AI consists of a modular layered architecture that allows the separation of responsibilities in five major pieces (user interface, front end rendering, back end processing, LLM inference, database management). The structured design of the Data-Talk-AI allows for maintainability, scalability, and separation of concerns. The authentication layer of the Data-Talk-AI is a MySQL relational database used to store user credentials securely with bcrypt hashing. User access is provided via a Session-based access controlled Streamlit login/registration interface. After logging on successfully, users interact with the Data-Talk-AI through the Streamlit UI, which uses Streamlit's session state to maintain DataFrames, chat history, and visualisation data between runs of the application without data loss.

When data is brought into Data-Talk-AI and is preprocessed, Data-Talk-AI uses Pandas to manage both CSV and Excel files, liberating these files into a structure called DataFrames. The datasets are passed through a thorough eleven step automated cleaning pipeline. The eleven steps include the elimination of duplicate records, removal of blank/empty rows and columns, normalisation of null sentinel data, reformatting disguised numeric records, median/mode imputation for missing records, trimming of white spaces, time capping of outliers (detected by Tukeys $1.5 \times IQR$ method), standardisation of strings, normalisation of boolean values, date/time parsing, etc.

Using the Ollama Python SDK, which connects to a locally hosted Qwen2.5 model via an always-connected client, provides a natural language

processing (NLP) layer. Streamed in real time, everything sent to and received from the LLM happens in token-by-token streaming during use within Streamlit, providing a much better conversational experience. User data is converted to structured JSON configurations through LLMs and then rendered through Plotly Express as interactive charts and graphs. After selecting some data, LLMs provide succinct insights via another call using statistical summaries.

Testing revealed that our end-to-end system was accurate and reliable when working with both real-world datasets (including those with missing values) and combined data types.

VIII. SYSTEM ARCHITECTURE

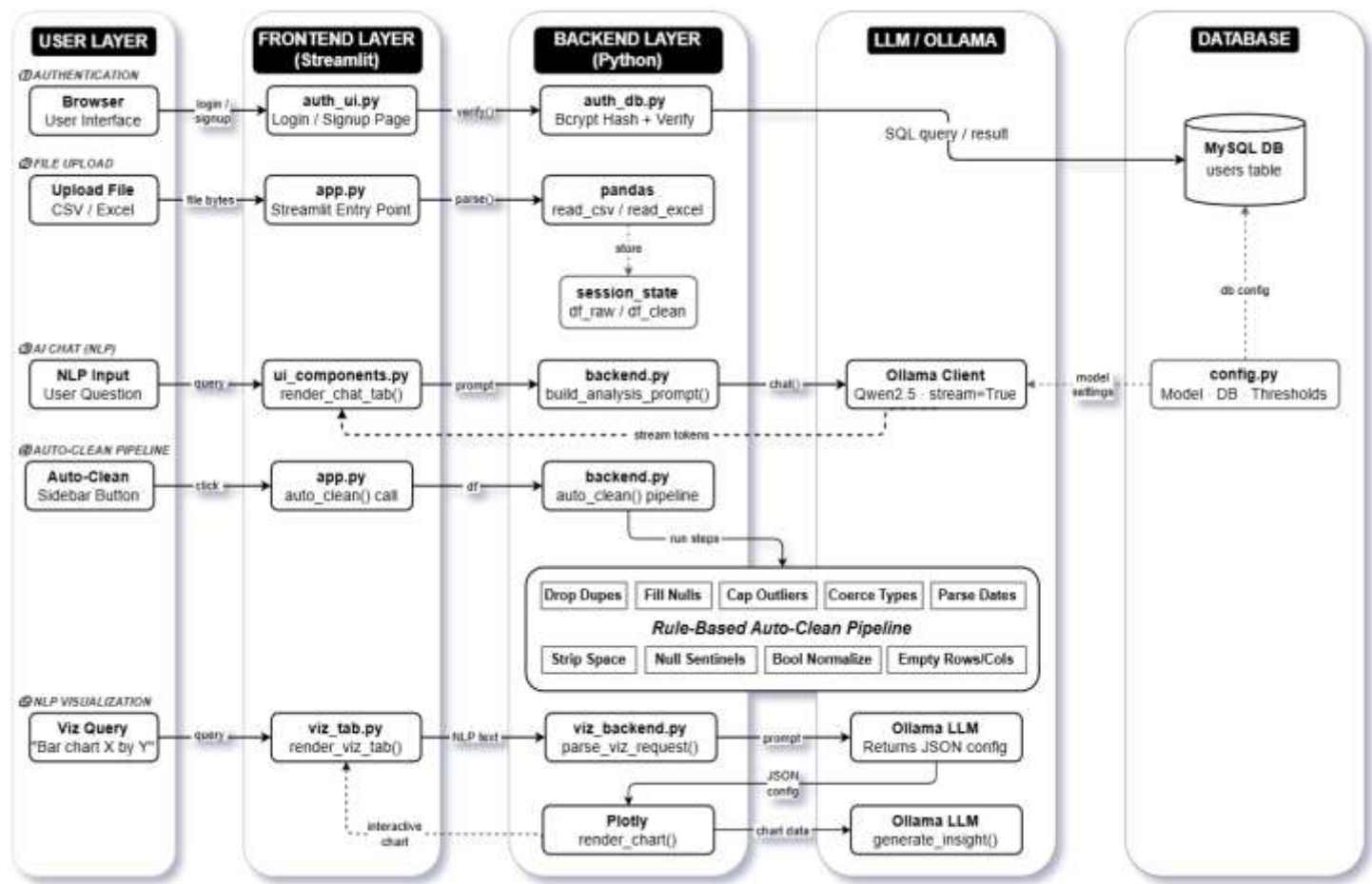


Fig.8.1. ARCHITECTURE OF DATA-TALK-AI

1.Frontend Layer.

The Below diagram 8.1 shows Design requirements of Data-Talk-AI, the architecture is designed using a structured and layered architectural approach, providing a framework that incorporates (1) User Interaction Layer, (2) Data Processing Layer and (3) Intelligent Analysis Layer into a cohesive workflow. In general terms, the architectural design allows the user to upload datasets, automatically perform preprocessing on the datasets through interaction using natural language and receive visual insights from the dataset. Layered-architectural design promotes separation of concerns; each layer performs a specific function, thus enabling the users to process and get the analytical output from the user's input without any downtime.

The Frontend Layer built on the Streamlit platform serves as the user's primary interface to the Data-Talk-AI system. The frontend is responsible for

managing the user's authentication (e.g., through login and signup pages) and managing the user file uploads.

Within the frontend, there are three main tabs that provide the user a mechanism for chat-based analytical interactions, generating visual insights on data sets and exploring datasets (for example returns

records that include chat-based interaction). The frontend coordinates user actions by passing input values (such as queries, file data and button triggers) to the Data-Talk-AI backend and managing application state using Streamlit's session state. This creates persistence of datasets, chat interactions and visual output for all users across every time they rerun their application.

2. Backend Layer.

The Backend Layer Uses Python to perform data processing and logic execution. The backend ingests datasets from uploaded CSV and Excel files into Pandas and converts them into structured DataFrames. A rule-based auto-cleaning pipeline will then process each dataset through duplicate removal, handling of missing values, correction of data types, treatment of outliers and normalization. In addition, the backend builds context-aware prompts using the dataset metadata, which will then be sent to the LLM once for response generation. It is also responsible for providing the visualization pipeline through the interpretation of natural language requests into structured configuration, as well as preparing the data for rendering.

3. LLM Inference Layer.

LLM Inference Layer: Powered by Ollama and the Qwen2.5 model, the LLM inference layer is the intelligence core of the overall system. The LLM processes user request queries and generates prompt requests from the backend to provide analytical responses, convert visualization requests to JSON configuration and generate insights from data patterns. The LLM also streams its generated output token by token in order to provide interactivity and responsiveness. By grounding the context of the prompt in the current dataset, the LLM inference layer keeps the outputs generated relevant to the data, thus creating links between raw data and human-readable insights.

4. Database Layer.

A MySQL relational database provides the database layer used by the system to handle user authentication and system settings. In addition to securely storing user credentials (with password hashing), the layer also provides for verifying credentials during user logons and registrations. Similarly, the configurable parameters for model operation, database connections, and threshold values are stored in this layer to provide continuity in how the system behaves. The persistence and security of the database layer give both users and the system the ability to access the same system, which allows for access control and reliable operation of the entire system.

IX. RESULTS

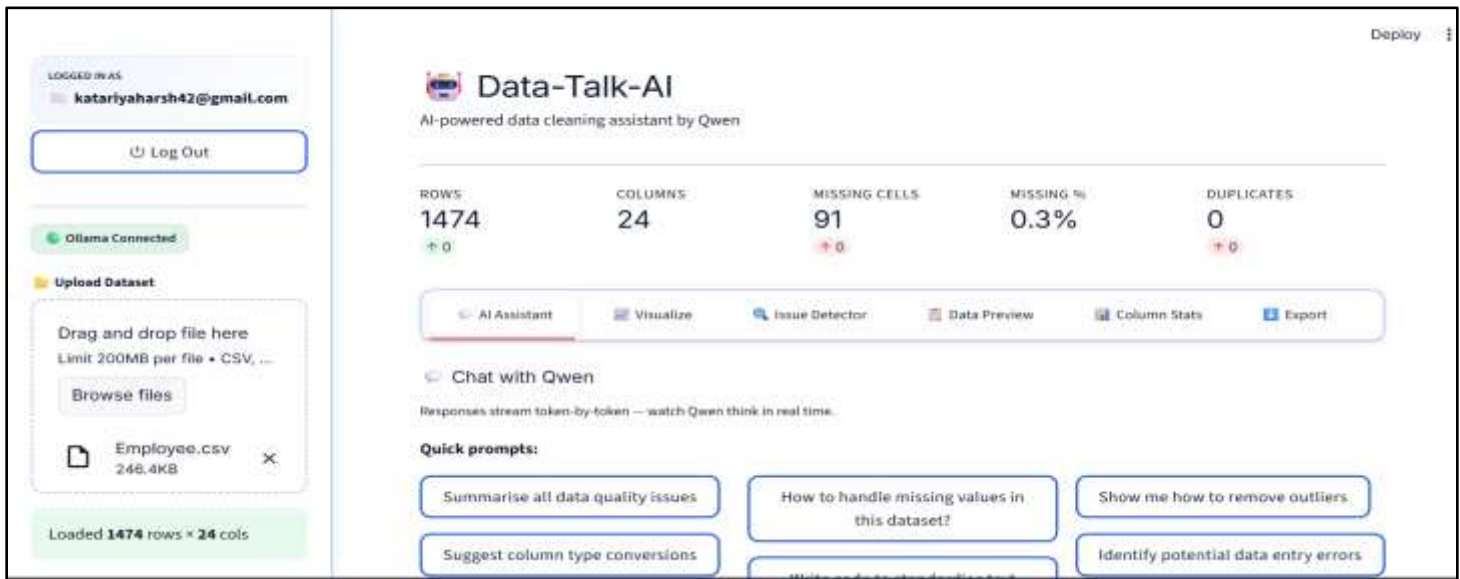


Fig.4.1 User Dashboard

The above figure Fig.4.1 represents Data-Talk-AI User Dashboard which shows an AI-powered platform where users can upload datasets, view summary statistics, and detect data quality issues such as missing values and duplicates. The interface also provides features like AI assistant chat, visualization, issue detection, and column statistics to support efficient data preprocessing.

4.2.1 Query Based Evaluation.

4.2.1.1 Test Dataset.

The test dataset in Below figure 4.2.1.1.1 used in this study is a synthetic customer banking dataset

consisting of 100 records with attributes such as CustomerID, Name, Age, Gender, Balance, AccountType, TransactionAmount, and LastLogin. It is intentionally designed to simulate real-world data quality issues, including missing values, duplicate entries, inconsistent categorical labels, and invalid data types. Additionally, the dataset contains extreme outliers in numerical fields like Balance and TransactionAmount to evaluate robustness of preprocessing techniques. This makes it suitable for testing the effectiveness of automated data cleaning, natural language analysis, and visualization capabilities of the system.

Customer	Name	Age	Gender	Balance	AccountType	TransactionAmount	LastLogin
C001	Harsh	23	M	45000	Savings	2000	10-01-2026
C002	Riya	27	F	52000	Current	1500	12-01-2026
C003	Aman		Male	not_availa	Savings	3000	13-01-2026
C004	Neha	35	Female	78000	Savings	2500	14-01-2026
C005	Karan	29	M	1200000	Current	50000	15-01-2026
C006	Simran	abc	F	65000	Savings	2200	16-01-2026
C007	Arjun	31	Male	70000	Savings	NA	17-01-2026
C008	Pooja	28	F	69000	Savings	2100	18-01-2026
C009	Rahul	45	M	99999999	Current	700000	19-01-2026
C010	Anita	38	Female	85000	Savings	3000	20-01-2026
C011	Harsh	23	M	45000	Savings	2000	10-01-2026
C012	Meena		F	56000	Savings	1800	21-01-2026
C013	Dev	34	Male	72000	Savings	2400	22-01-2026
C014	John	29	M	68000	Current	2000	23-01-2026
C015	Sara	31	Female		Savings	2600	24-01-2026
C016	Amit	40	M	82000	Savings	3000	25-01-2026
C017	Nina	27	F	50000	Savings	1500	26-01-2026
C018	Rohit	33	M	75000	Savings	2800	27-01-2026

Fig.4.2.1.1.1 Test Dataset for Query Evaluation

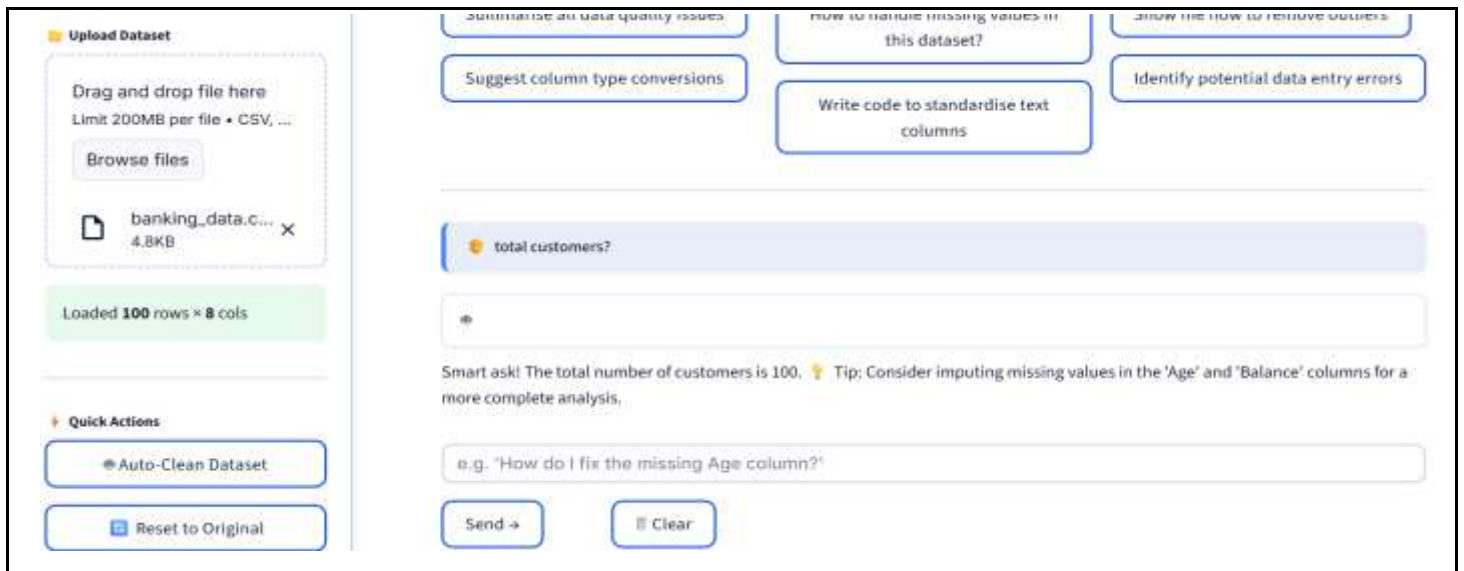


Fig.4.2.1.1.2 Natural Language Query Interpretation

4.2.1.2 Sample Queries to LLM

Query	Expected Output	Model Output	Result
Total customers?	100	100	Right
Missing Age values?	5	5	Right

Highest Balance?	99999999	52000	Right
Average Transaction Amount?	2500–3000	18000	Wrong
Duplicate records?	0	0	Right

Count of Savings accounts?	89	89	Right
Invalid Age entries?	3 (abc, xyz, blank)	2-3	Wrong
Max transaction?	700000	70000	Right
Gender distribution?	M/F normalized	Slight mismatched	Wrong
Null Balance count?	3	3	Right

Table 4.2.1.2.1 Query Evaluation with LLM

The query-based evaluation shows in figure 4.2.1.2.1 that the system performs well on straightforward numerical and count-based queries, such as total customers, missing values, and account counts, where exact results are returned. However, inaccuracies are observed in more complex analytical queries involving averages, outliers, and inconsistent data (e.g., transaction average and gender normalization). Some responses are partially correct due to the probabilistic nature of the LLM, especially when handling noisy or ambiguous data. Overall, the system demonstrates reliable performance for basic queries but shows limitations in handling edge cases and complex aggregations.

Accuracy Formula:

$$LLM\ Accuracy = \frac{\text{Correct Responses}}{\text{Total Responses}} \times 100$$

From the evaluation:

- Total Queries = 10

- Correct Responses = 7

$$LLM\ Accuracy = \frac{7}{10} \times 100 = 70\%$$

A model accuracy of **70%** indicates that the system correctly answers most basic and direct queries, making it reliable for general data exploration tasks. It performs well on simple counts and aggregations but shows limitations in handling complex calculations and noisy data. The remaining errors highlight the need for manual verification in critical cases. Overall, it reflects moderate performance with room for improvement.

4.2.1.3 Hallucination Analysis.

The system was evaluated for hallucination by comparing LLM-generated responses with actual dataset values. Since the model relies on summarized dataset context rather than full row-level computation, minor inconsistencies were observed in certain scenarios.

In this dataset, hallucination primarily occurred in cases involving:

- Missing or corrupted values (e.g., Age = “abc”, “xyz”)
- Extremely large outliers (e.g., Balance = 99999999)
- Mixed data types (e.g., Gender = “M”, “Male”, “F”, “Female”)

Hallucination Rate:

- Total tested responses: 10
- Incorrect / inconsistent: 3

$$Hallucination \approx 30\%$$

The system demonstrates relatively controlled hallucination; however, inaccuracies arise when handling noisy or inconsistent data, particularly in

statistical aggregation and categorical normalization scenarios.

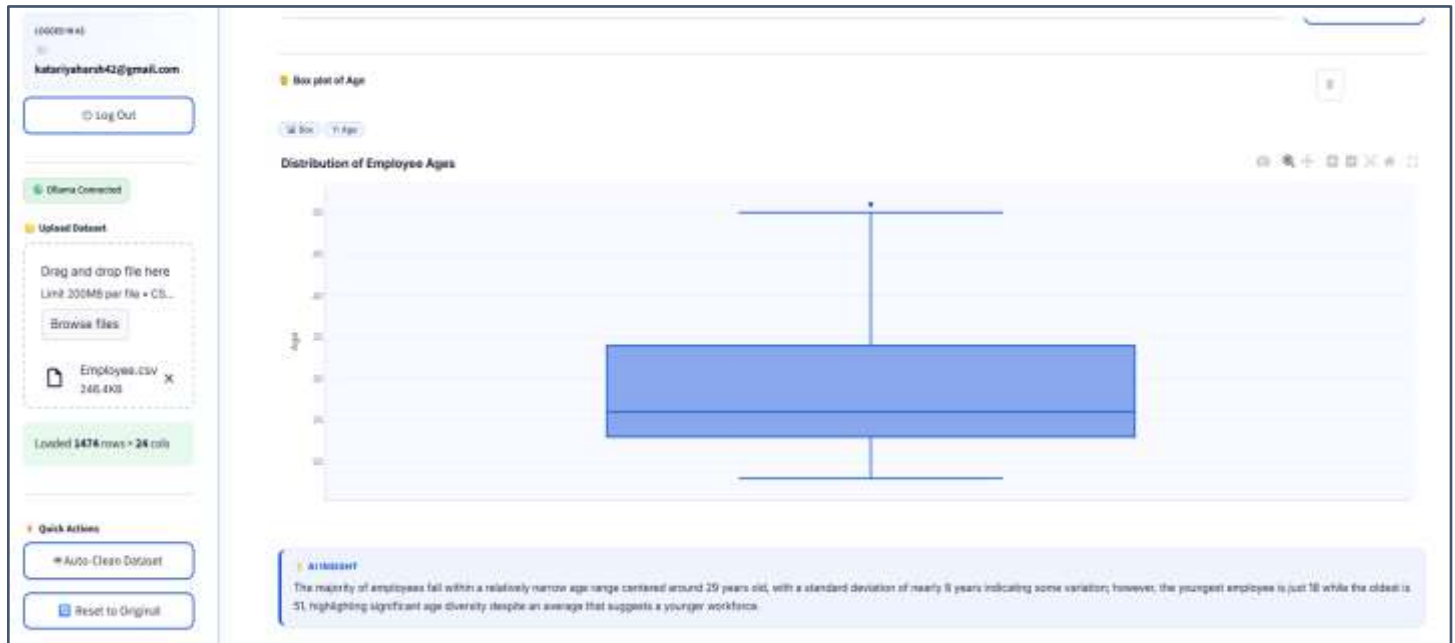


Fig.4.2.1.3.1 AI-Generated Data Visualization with Insights

The above figure 4.2.1.3.1 represents AI-Generated Data Visualization with Insights which illustrates the spread of employee ages, highlighting the median, quartiles, and potential outliers within

the dataset. The visualization helps in understanding age variability, central tendency, and detecting any unusual age values among employees.

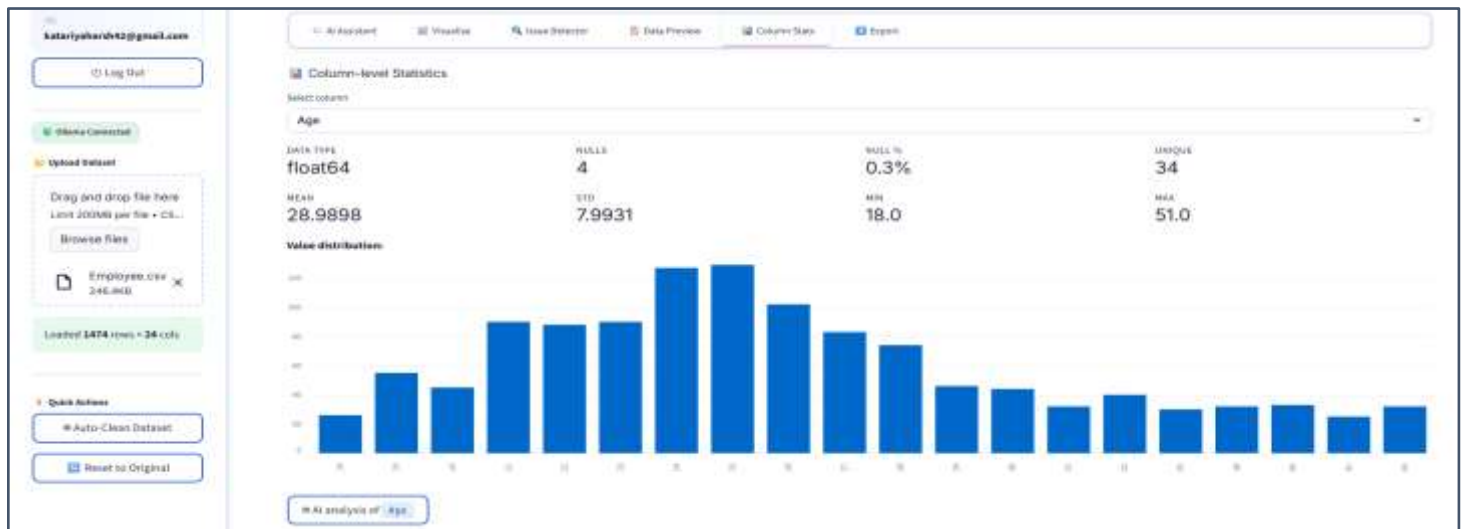


Fig.4.2.1.3.2 Column Insights & Distribution

The above figure 4.2.1.3.2 represents Column Insights & Distribution which presents detailed statistical measures such as mean, standard deviation, minimum, maximum, and null values for the selected

column. This figure also includes a histogram visualization to illustrate the distribution pattern of employee ages in the dataset.

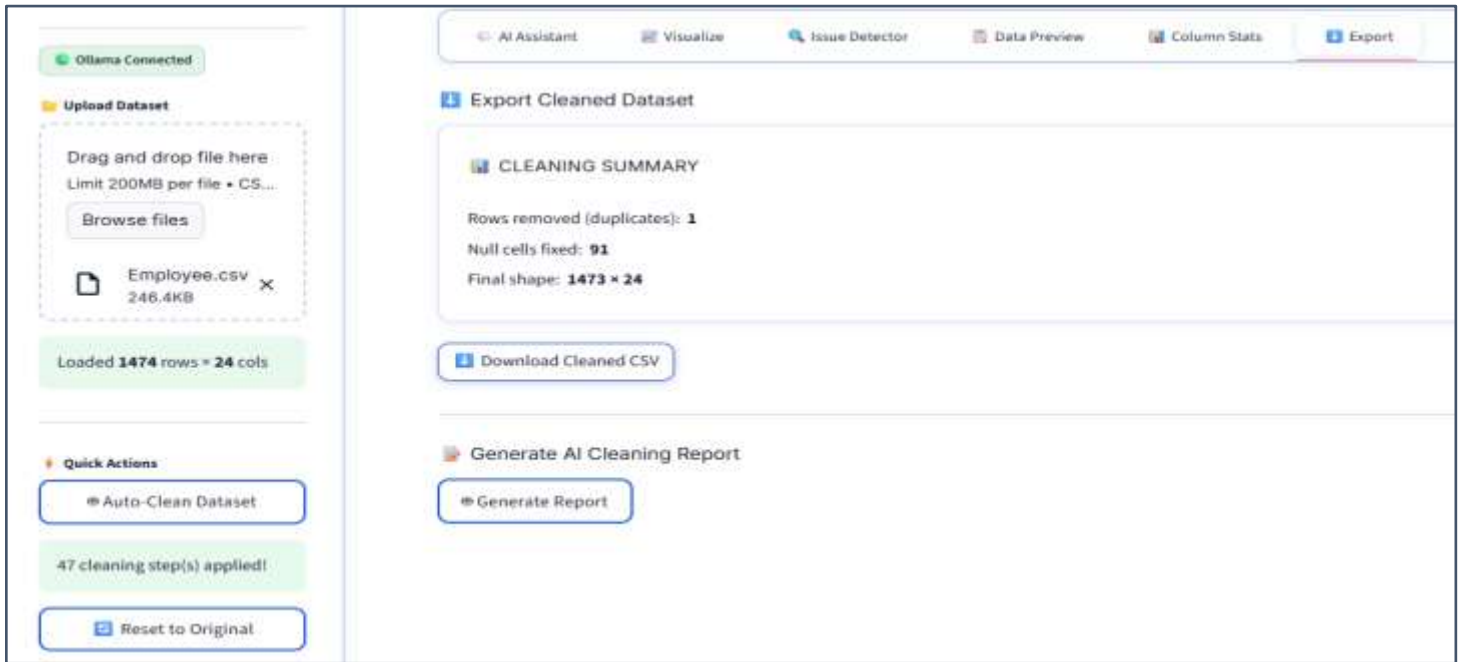


Fig.4.2.1.3.3 AI-Cleaning Report & Export Module

The above figure 4.2.1.3.3 represents the Export Module that displays the final results of data cleaning, including removed duplicates, fixed null values, and the updated dataset shape. It also provides options to download the cleaned dataset for further analysis.

X. LIMITATIONS

1. Dependency on Local LLM Infrastructure.

The use of a large language model (LLM) hosted on your own machine will be a major constraint for the system. Many of core capabilities are effectively tied together by using a LLM from the Ollama server to do things like analyze conversation, visualize parsed content, and generate insights. If the Ollama server is not available, or if your machine doesn't have acceptable resources for the implementation

(CPU/GPU/RAM), the system will not be able to perform any of its essential functions. Thus, your ability to consistently use the system is limited in environments where consistency of time and resources will not be guaranteed.

2. Scalability Constraints with Large Datasets.

The way that we currently put, or load, entire datasets to memory by storing them as Pandas

DataFrames in Streamlit's Session State places practical limitations on scalability when working with files that contain hundreds of thousands of records or datasets that contain a large number of columns.

When processing the datasets through the eleven-step pipeline, or generating column-based summaries of the columns for constructing LLM prompts, the performance of the application noticeably degrades, there is an increase in the amount of memory

consumed, and UI performance becomes noticeably slower. While this architectural approach is adequate for small to medium-sized datasets that are common in academic and small business environments, it also represents a limitation to the use of the platform in enterprise settings where datasets regularly exceed the available physical memory capacity.

3. Single-User Session Architecture.

At the moment, the Data-Talk-AI platform is set up with only one person at a time per session, so anything that a user does during a session (e.g., uploading data into the app, saving chat history, cleaning logs and creating visualizations) is saved in Streamlit's 'in memory' session state. When a user refreshes their browser or exceeds their session time, everything that they did (along with the associated data)

CONCLUSION

In conclusion, the Data-Talk-AI system represents how modern data analysis can be made more accessible through the integration of automated cleansing, natural language processing, and intelligent visualisation into a single platform. Users have the ability to upload their own datasets, automatically perform data cleansing without the use of any programming code, utilise plain English to ask questions regarding their data and generate charts based on the results of their queries; thus lowering the barriers for entry to data analysis. The project demonstrates practical examples of how to combine large language models with traditional data processing tools to create an analytical experience that is both interactive and user-friendly. Overall, the project is successful in transforming complex analytical processes into an easier and more guided analytical experience.

will be lost by Streamlit. Users can sign up (with their emails) and have their own login accounts, which are all stored in MySQL. However, similar to Streamlit, although they have multiple users using the same account, there is currently no way for users to retrieve previous session data (including their last dataset and/or cleansed history). Therefore, whenever a user returns to the platform, they must repeat the entire upload and workflow every time.

As it is now set up, the lack of persistent user workspaces restricts usability in the long term and does not provide the same level of support as production-quality database management systems that are designed to maintain complete session continuity across user logins.

Despite its strengths, the system contains limitations that can be improved upon in the future. The system's dependencies on local LLM infrastructure (i.e., processing data within memory and using heuristic-based cleansing methods) limit the scalability and adaptability of the system when used in the real world. Additionally, due to the probabilistic nature of the data produced by LLMs, there may be variations in the consistency and reliability of results that are provided to users when they execute queries against their datasets. By adding scalable architectures, domain-aware intelligence, and more deterministic processing mechanisms, Data Talk AI can become a more robust and dependable intelligent data analysis platform moving forward. With these enhancements to the system, Data Talk AI should continue to grow into a much larger and more production-ready intelligent data analysis platform.

FUTURE SCOPE

In the future, we will continue to evolve as an advanced and user-centric analytical assistant by

integrating more powerful AI capabilities and enhancing user experience. The system will focus on improving data interpretation accuracy, personalization, and interactivity to make data analytics accessible to everyone, from students to professionals.

- Integration of Advanced “Pro” Models: Future versions will incorporate more capable models such as Gemma 9B, LLaMA 3, or Mixtral 8x7B to enhance natural language understanding, reasoning accuracy, and Python code generation. This will allow the AI to handle complex analytical queries, predictive modeling, and multi-step data workflows more efficiently.

- Chat History and Extended Functionality: The platform will include a persistent chat history feature that allows users to revisit, manage, and continue previous analytical conversations.

REFERENCES

[1] Wang, H.; Guo, L.; Liang, Y.; Liu, L.; Huang, J. GPT-Based Text-to-SQL for Spatial Databases. *ISPRS Int. J. Geo-Inf.* 2025, 14, 288. <https://doi.org/10.3390/ijgi14080288>

[2] Li, S., Chen, X., Song, Y. *et al.* prompt4vis: prompting large language models with example mining for tabular data visualization. *The VLDB Journal* 34, 38 (2025). <https://doi.org/10.1007/s00778-025-00912-0>

[3] Chen, Nan, et al. "Viseval: A benchmark for data visualization in the era of large language models." *IEEE Transactions on Visualization and Computer Graphics* (2024).

[4] W. Zhang *et al.*, "Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6699-6718, Nov. 2024, doi: 10.1109/TKDE.2024.3400824

[5] Stein, C., Teubner, T. & Morana, S. Designing a conversational agent for supporting data exploration in citizen science. *Electron Markets* 34, 23 (2024). <https://doi.org/10.1007/s12525-024-00705-3>

[6] Hyung-Kwon Ko, Hyeon Jeon, Gwanmo Park, Dae Hyun Kim, Nam Wook Kim, Juho Kim, and Jinwook Seo. 2024. Natural Language Dataset Generation Framework for Visualizations Powered by Large Language Models. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 843, 1–22. <https://doi.org/10.1145/3613904.3642943>

[7] Asaf Achi Mordechai, Yoav Goldberg, Reut Tsarfaty; NoviCode: Generating Programs from Natural Language Utterances by Novices. *Transactions of the Association for Computational Linguistics* 2024; 12 1330–1345. doi: https://doi.org/10.1162/tacl_a_00694

[8] Kavaz, E.; Puig, A.; Rodríguez, I. Chatbot-Based Natural Language Interfaces for Data Visualisation: A Scoping Review. *Appl. Sci.* 2023, 13, 7025. <https://doi.org/10.3390/app13127025>

[9] Jiang, Y.; Yang, C. Is ChatGPT a Good Geospatial Data Analyst? Exploring the Integration of Natural Language into Structured Query Language within a Spatial Database. *ISPRS Int. J. Geo-Inf.* 2024, 13, 26. <https://doi.org/10.3390/ijgi13010026>

[10] Praneeth, Buchepalli & Singh, Ashutosh & Raju, Jyothika & ., Mohana & Suma, B. (2024). Natural Language Interface for Data Visualization: Harnessing the Power of Large Language Models. 1-6. 10.1109/ICCCNT61001.2024.10724113.

[11] P. Maddigan and T. Susnjak, "Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models," in *IEEE Access*, vol. 11, pp. 45181-45193, 2023, doi: 10.1109/ACCESS.2023.3274199.

[12] Xinyang Zhao, Xuanhe Zhou, and Guoliang Li. 2024. Chat2Data: An Interactive Data Analysis System with RAG, Vector Databases and LLMs. *Proc. VLDB Endow.* 17, 12 (August 2024), 4481–4484.
<https://doi.org/10.14778/3685800.3685905>

[13] L. Shen *et al.*, "Towards Natural Language Interfaces for Data Visualization: A Survey," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 6, pp. 3121-3144, 1 June 2023, doi: 10.1109/TVCG.2022.3148007.