World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

Check for updates

A Streamlet based interactive system for LLM assisted SQL generation and chart rendering

Bhargavi Jangam, Snehalatha Chevula *, Vennela Koorella and Hithesh Mailarapu

Department of CSE (AI and ML), ACE Engineering College, India.

World Journal of Advanced Research and Reviews, 2026, 29(03), 1386-1393

Publication history: Received on 09 February 2026; revised on 16 March 2026; accepted on 18 March 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.29.3.0677

Abstract

The project seeks to create an interactive system that facilitates easy navigation and analysis of data. Instead of requiring the user to write complex queries or having the necessary technical knowledge, the system allows users to interact with data using natural language queries. Users can upload their data and then query it in a simple way, such as comparing, summarizing, or graphing the data. The system will then process the queries and display the data in the form of tables, graphs, or charts.

The project makes data analysis more accessible by allowing users to find patterns, trends, and insights in the data without requiring specialized knowledge. The project bridges the gap between data and meaningful information, allowing organizations and individuals to make informed decisions based on data. The project makes it possible for an intuitive and user-friendly interface to revolutionize the way people interact with data, making it a seamless and interactive process rather than a technical one.

Keywords: Conversational Data Interface; Natural Language Processing (NLP); Data Visualization; Large Language Models (LLMS)

1. Introduction

Data has become a new form of value in the current digital era, but the data analysis process is still not accessible to non-technical people due to the requirement of SQL knowledge and visualization skills, thus creating an “analysis gap” in data-driven decision-making. This research project, titled “A Streamlet-Based Interactive System for LLM-Assisted SQL Generation and Chart Rendering,” seeks to address this issue by leveraging Large Language Models (LLMs) to enable natural language-based data exploration. The proposed system will allow users to upload their data and ask questions in natural English, which will be automatically translated into optimized SQL queries using advanced prompt engineering and in-context learning. After the execution of the query, the system will automatically analyze the data and generate corresponding visualizations, such as charts or graphs, using a Streamlet-based interface. By seamlessly integrating Text-to-SQL and Text-to-Visualization capabilities, this research project seeks to make data analysis processes a real-time, interactive, and intuitive experience.

* Corresponding author: Snehalatha Chevula

2. Literature Review

2.1. Pushpendu Ghosh, Aryan Jain, Promod Venigalla. SQL Genie: A Practical LLM based System for Reliable and Efficient SQL Generation (2025)

The architecture of this system consists of three components: a Table Onboarded for pre- processing the schema, a hybrid LLM-based SQL Generator, and a Feedback Augmentation loop. The SQL generator initially attempts to match

queries with a set of tested examples before resorting to a multi-agent pipeline (Planner, Generator, Critic) for new queries. The results of this research reveal that the hybrid system is very efficient, with a 64% reduction in SQL generation time compared to traditional multi-LLM approaches, and that Auto Correctors for post- processing improve query success rates in a production environment.

2.2. Victor Dibia. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models (2023)

LIDA employs a multi-stage LLM pipeline to automate visualization. It begins with a SUMMARIZER module that creates a natural language summary of the data, followed by a GOAL EXPLORER that suggests potential visualization goals. The VISGENERATOR module then takes a goal and generates "grammar-agnostic" visualization code for various libraries (e.g., Matplotlib, Seaborn). An optional INFOGRAPHER module can transform standard charts into stylized infographics. A key finding is that the system can automatically discover meaningful visualization goals from a dataset, which is a powerful feature for data exploration.

2.3. YANG WU, YAO WAN, HONGYU ZHANG, YULEI SUI, WUCAI WEI, WEI ZHAO, GUANDONG XU, HAI JIN

2.3.1. Automated Data Visualization from Natural Language via Large Language Models: An Exploratory Study (2024)

This paper conducts an empirical study to find the most effective methods for using LLMs in Natural Language to Visualization (NL2VIS) tasks. It systematically tests various strategies for transforming tabular data into sequential text prompts and investigates iterative optimization techniques like Chain-of-Thought (COT) and self-repair to correct LLM output. The study concludes that representing tabular data in a programmatic format (like a SQL CREATE TABLE statement) is a highly effective prompting strategy. It also confirms that iterative update strategies, such as self- repair, are effective at correcting initial generation errors.

2.4. Weixu Zhang, Yifei Wang, Yuanfeng Song, Victor Junqiu Wei, Yuxing Tian, Yiyan Qi, Jonathan H. Chan, Raymond Chi-Wing Wong, and Haiqin Yang. Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey (2024)

This paper provides a comprehensive survey of the Natural Language Interface (NLI) field for both Text-to-SQL and Text-to-Visualization (Text-to-Vis) tasks. It categorizes the evolution of research into three main stages: Traditional (rule-based), Neural Network (sequence-to-sequence), and Foundation Model (LLM-based). The review shows that modern LLMs have completely changed the game, making these tools much more powerful and flexible. A key insight is that Text-to-SQL and Text-to-Vis are usually worked on as two separate problems, suggesting a need for more tools that do both together seamlessly.

2.5. Comparison Table

Table 1 Summary of Key Finding in recent literature

Paper	Authors	Methodology	Contribution	Limitations
SQL Genie: A Practical LLM- based System for Reliable and Efficient SQL Generation (2025)	Pushpendu Ghosh, Aryan Jain, Promod Venigalla	Three-component architecture: Table Onboarded (schema preprocessing), Hybrid LLM SQL Generator (example matching + multi-agent Planner–Generator–Critic pipeline), Feedback Augmentation with Auto Correctors	Achieved 64% reduction in SQL generation time; improved reliability and production-level query success through iterative refinement	Primarily focused on SQL generation; does not deeply integrate visualization; system complexity may increase computational overhead

LIDA: Automatic Generation of Grammar-Agnostic Visualizations using LLMs (2023)	Victor Dibia	Three-component architecture: Table Onboarded (schema preprocessing), Hybrid LLM SQL Generator (Example matching + multi-agent Planner–Generator–Critic pipeline), Feedback Augmentation with Auto Correctors	Achieved64% reduction in SQL generation time; improved reliability and production-level query success through iterative refinement	Primarily focused on SQL generation; does not deeply integrate visualization; system complexity may increase computational overhead
Automated Data Visualization from Natural Languagevia LLMs: An Exploratory Study (2024)	Yang Wu, Yao Wan, Hongyu Zhang, Yulee Sui, Wutai Wei, Wei Zhao, Guandong Xu, Hai Jin	Empirical evaluation of NL2VIS prompting strategies; tests schema representation (CREATE TABLE format), Chain-of- Thought (COT), and self-repair mechanisms	Demonstrates effectiveness of schema-aware promptingand iterative correction forimproving visualization accuracy	Exploratory and experimental in nature; does not propose a full end-to-end deployed system
Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey (2024)	Weixi Zhang, Yifei Wang, Yanfeng Song, Victor Jonquil Wei, Yixing Tian, Yayan Qi, Jonathan H. Chan, Raymond Chi-Wing Wong, Haiqing Yang	Comprehensive survey categorizing NLI evolution: Rule- based → Neural Networks → Foundation Models (LLMs)	Identifies research trends, challenges, and gap between Text-to-SQL and Text-to- Visualization systems	Survey paper; does not implement or experimentally validate a unified system

2.6. Research Gaps

The existing systems in this domain show strong but disjointed capabilities. SQLGenie is a powerful backend system focused solely on the generation of accurate and efficient SQL queries. However, the system does not extend its workflow to the visualization of the data. LIDA is a comprehensive system for generating visualizations from structured data sources such as CSV files. However, the system assumes the data is already clean and does not focus on the difficult task of data extraction and aggregation through the generation of accurate and efficient SQL queries. Other academic works offer useful techniques in the domain of prompt engineering and automated error correction for large language models. However, these works are not presented in a unified and user-friendly way. Finally, the results from the survey show that the fields of Text-to-SQL and Text-to-Visualization are treated separately in the literature, indicating a significant gap in the development of unified systems in the field.

3. Proposed Methodology

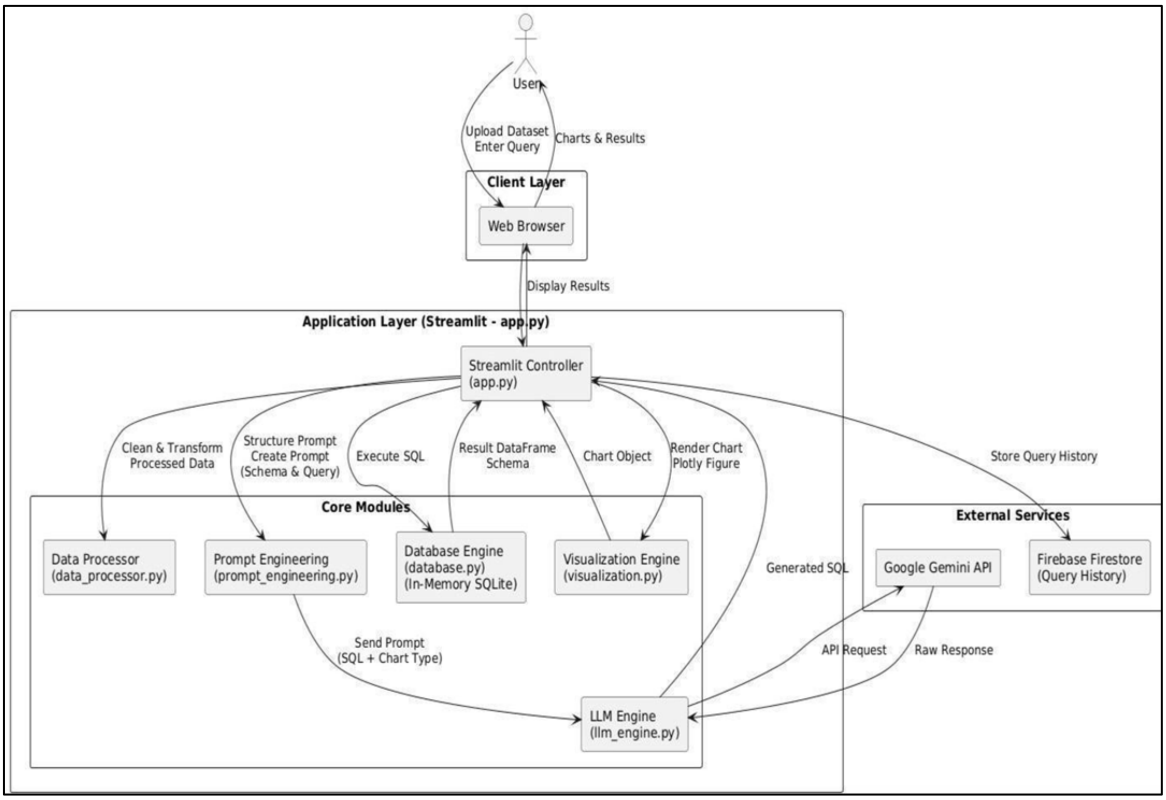

Figure 1 System architecture diagram

3.1. Algorithm

- Initialize the Streamlit application and required modules (Database Engine, Prompt Template, Leonine, Data Processor, Visualization Engine).
- Upload the dataset using Streamlit's file uploader and read it into a Pandas DataFrame.
- Load the dataset into an in-memory SQLite database and extract the table schema.
- Accept the user's natural language query through the input interface.
- Createa structured prompt by combining the extracted schema, userquery, and SQL generation rules.
- Send the prompt to the Gemini API via the LLMEngine.
- Extract and validate the JSON response from the LLM output.
- Retrievethe SQL query and chart metadata (chart type and axis mapping).
- Validate that the generated SQL contains only safe SELECT operations.
- Execute the SQL query on the SQLite database and obtain the result as a DataFrame.
- Ifthe result is empty, display a "No Results Found" message.
- Otherwise, process the result using the DataProcessor (data type conversion, missing value handling, sorting, row limiting).
- Generate the appropriate visualization using the VisualizationEngine and Plotly.
- If the "Display SQL" option is selected, display the SQL query along with the chart and raw result data; otherwise, display only the chart and result data.
- Storethe query history in Firebase (if the user is authenticated).
- Maintain workflow continuity using Streamlit session state.

3.2. Output

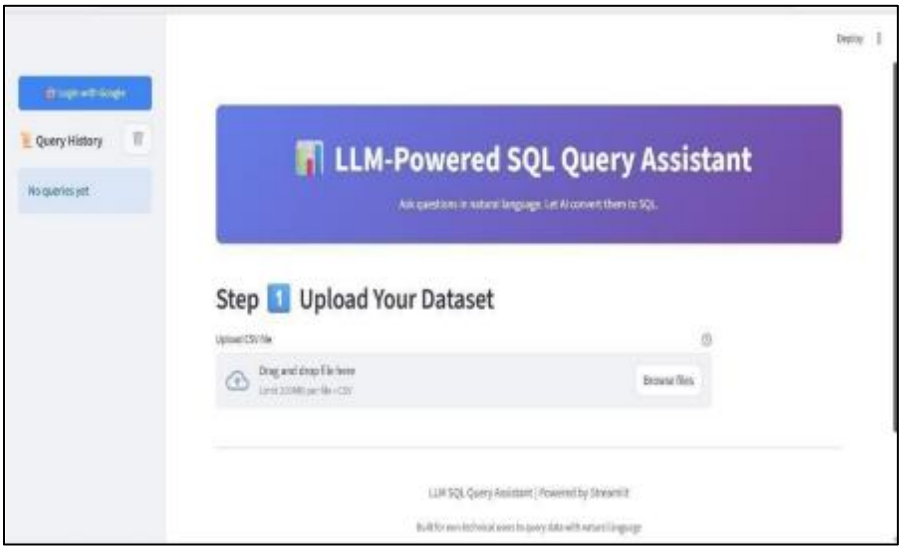

Figure 2 System Home Screen

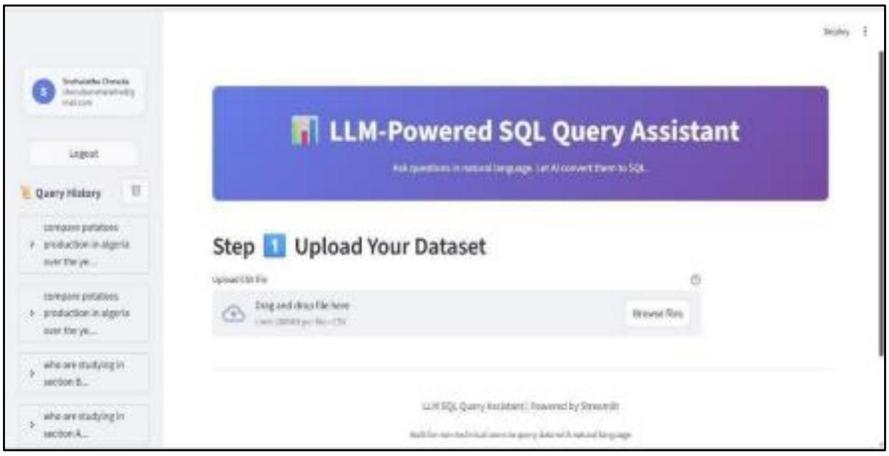

Figure 3 Dataset Upload Screen

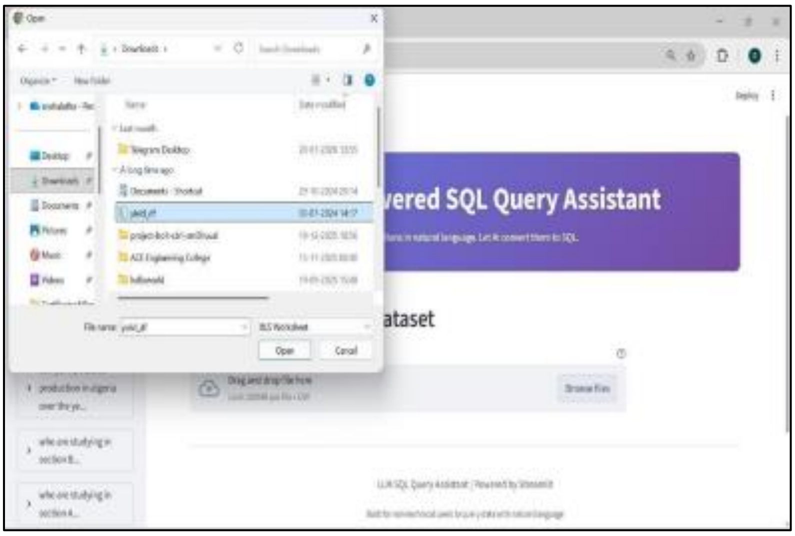

Figure 4 File Selection Window

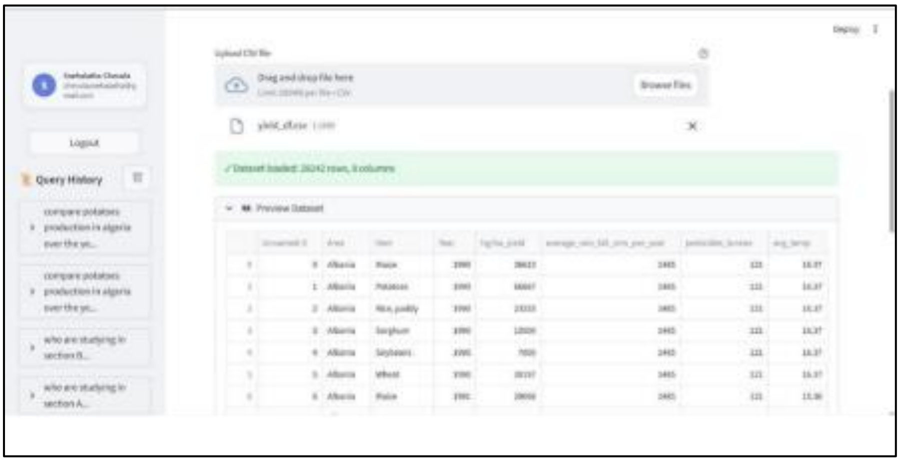

Figure 5 Dataset Preview Screen

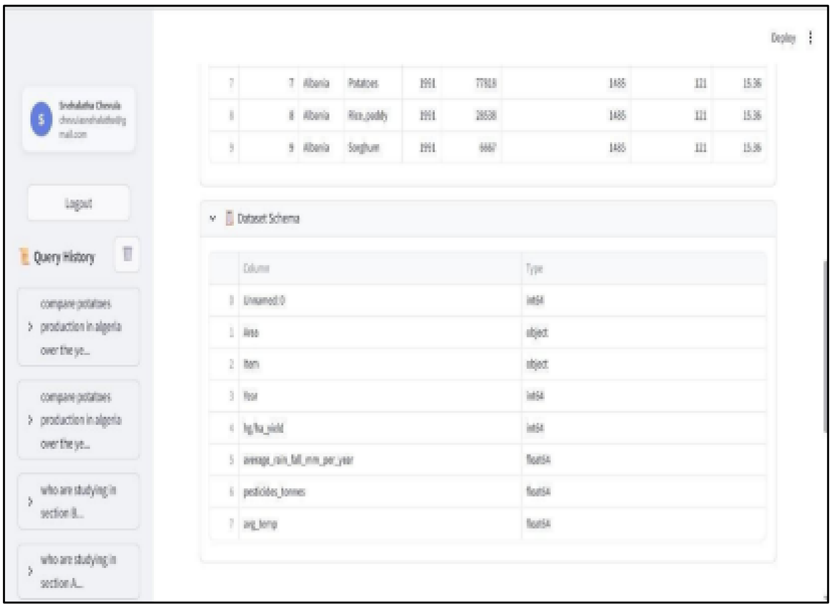

Figure 6 Data Summary Screen

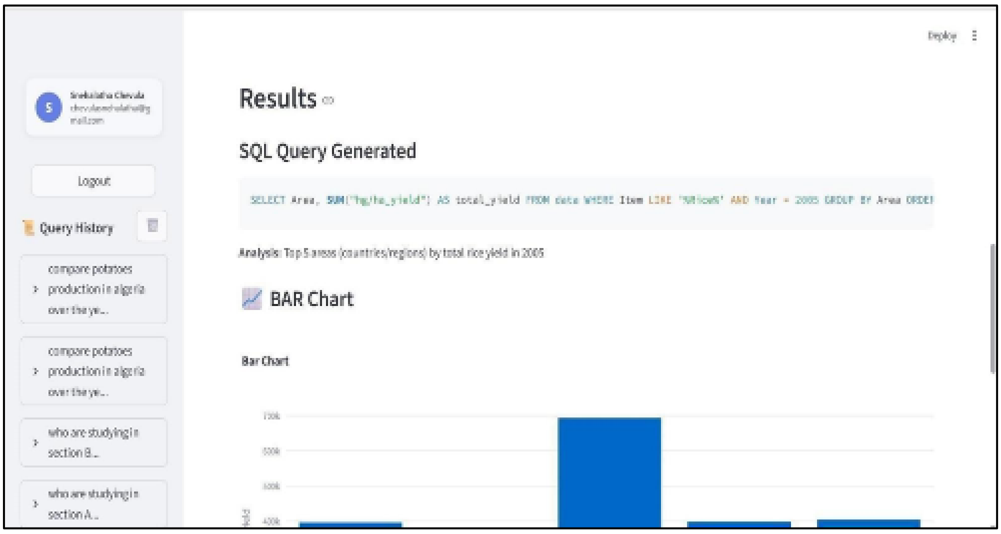

Figure 7 Query Input Screen

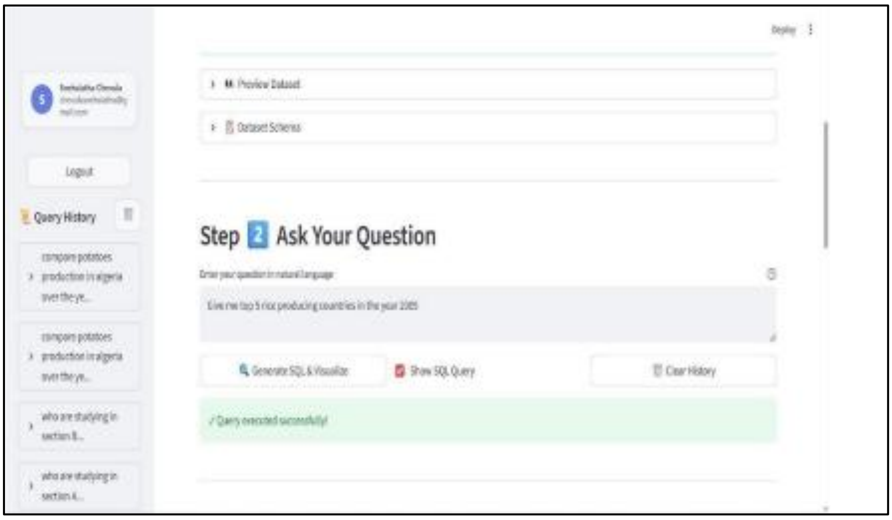

Figure 8 SQL Query Result Screen

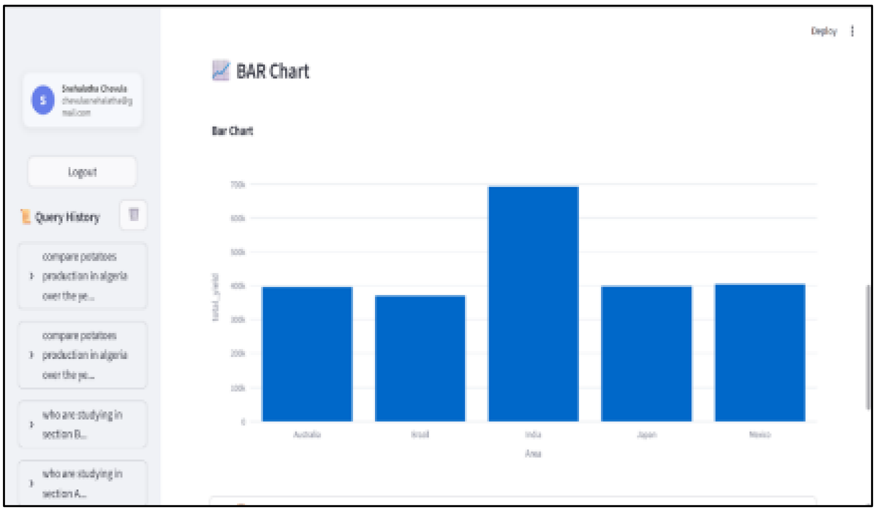

Figure 9 Chart Visualization Screen

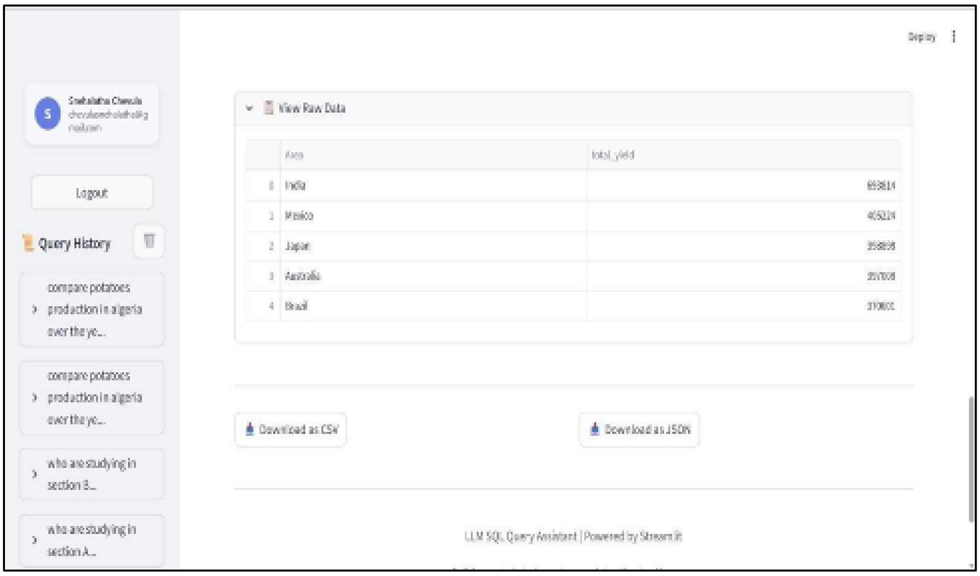

Figure 10 Raw Data Output Screen

4. Conclusion

The proposed project has successfully offered a comprehensive and integrated solution that bridges the gap between text-to-SQL generation and automated data visualization. The proposed system has significantly improved upon existing approaches that only deal with these components as standalone entities. Instead, our proposed framework has offered a seamless and integrated system that can transform text-based queries into optimized SQL queries, extract structured data from relational databases, and display insightful visualizations through a single interface. The proposed system has used advanced prompt engineering, query validation mechanisms, and dynamic visualization components that can offer robustness, accuracy, and operational efficiency. The proposed system has offered an efficient and scalable system that can significantly contribute to the development of AI-based intelligent data exploration tools.

Compliance with ethical standards

Disclosure of conflict of interest

No conflicts of interest to be disclosed.

References

[1] P. Ghosh, A. Jain, and P. Yenigalla, “SQLGenie: A Practical LLM-Based System for Reliable and Efficient SQL

[2] Generation,” arXiv preprint arXiv, 2023.

[3] V. Dibia, “LIDA: Automatic Generation of Visualizations and Infographics Using Large Language Models,” arXiv preprint arXiv, 2023.

[4] Y. Wu, Y. Wan, H. Zhang, Y. Sui, W. Wei, W. Zhao, G. Xu, and H. Jin, “Automated Data Visualization from Natural Language via Large Language Models,” in Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE), 2023.

[5] W. Zhang, Y. Wang, Y. Song, V. J. Wei, Y. Tian, Y. Qi, J. H. Chan, R. C. Wong, and H. Yang, “Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey,” ACM Computing Surveys,2023