

A conversational data analysis system using large language model

Wumi AJAYI *, Ihuanyachi O. OGBONNA and Daniel I. OGHINAN

Department of Software Engineering, Babcock University, Ilishan Remo, Ogun State, Nigeria.

Global Journal of Engineering and Technology Advances, 2026, 28(01), 027–035

Publication history: Received on 25 May 2026; revised on 30 June 2026; accepted on 02 July 2026

Article DOI: <https://doi.org/10.30574/gjeta.2026.28.1.0166>

Abstract

In today's data-driven environment, the ability to explore and manipulate datasets is essential, however, many existing data analysis tools require technical expertise in platforms such as Excel, Python, or Power BI, creating accessibility barriers for non-technical users. This project presents Quiksight, a web-based conversational data analysis system powered by Large Language Models (LLMs) that enables users to interact with datasets using natural language. The objective of this work was to develop an intuitive system that simplifies data analysis by allowing users to upload Excel or CSV files and perform operations such as filtering, sorting, summarization, column modification, missing-value handling, and data export without writing code. The system was implemented using Google Gemini as the LLM, FastAPI for backend processing, and HTML, JavaScript, and Tailwind CSS for the frontend interface. Evaluation showed that Quiksight successfully interpreted and executed a wide range of natural language data manipulation queries with high accuracy. Testing recorded an average query response time of approximately 2 seconds, while dataset uploads up to 30MB remained below 20 seconds under suitable network conditions. Compared with existing platforms such as Julius.ai and PowerDrill.ai, Quiksight demonstrated advantages through its lightweight architecture, simplified interaction model, and focus on everyday users. The project demonstrates the feasibility of LLM-powered conversational interfaces for democratizing data analysis. Future improvements include database integration, support for larger datasets, and advanced analytical capabilities such as statistical modelling.

Keywords: Large Language Models (LLMs); Conversational AI; Data Analysis; Natural Language Processing (NLP); Data Democratization

1. Introduction

In the contemporary data-driven landscape, structured data formats such as CSV and Excel remain fundamental to organizational decision-making, scientific research, and operational planning [1]. The ability to efficiently explore and manipulate such datasets is therefore essential. However, traditional data analysis workflows often require programming knowledge or advanced spreadsheet skills, which limits accessibility for non-technical users [2] [3]. Recent advancements in Artificial Intelligence (AI), particularly Large Language Models (LLMs), have significantly transformed human-computer interaction by enabling natural language interfaces. These models allow users to interact with systems using everyday language rather than formal query languages or programming syntax, thereby reducing technical barriers to data analysis [4] [5].

LLMs such as GPT-style architectures have demonstrated strong performance in language understanding and code generation, enabling new paradigms for conversational analytics and automated data processing [6]. More recent surveys highlight their growing role in data management tasks, including summarization, transformation, and reasoning over structured datasets [5] [7]. Despite these advances, many existing systems remain oriented toward technical users, exposing code outputs or requiring an understanding of underlying analytical processes. This creates a

* Corresponding author: Wumi AJAYI.

need for lightweight, conversational systems that allow users to perform common data operations without programming expertise[8] .

This work introduces Quiksight, a web-based conversational data analysis system that integrates LLMs with traditional data processing frameworks. The system enables users to upload datasets, interact through natural language, perform transformations, and export results without writing code. The objectives of this research are: (1) to develop a web-based platform for dataset upload and automated summarization; (2) to implement a conversational interface for data manipulation; and (3) to evaluate usability and performance compared to existing conversational analytics tools. This work contributes to the broader field of conversational AI by demonstrating how natural language interfaces can reduce barriers to structured data analysis.

The remainder of this paper is structured as follows: Section 2 reviews pertinent literature on conversational AI and LLM-based analysis; Section 3 details the methodology and system architecture; Section 4 presents the evaluation results; and Section 5 offers concluding remarks and directions for future research.

2. Literature Review

Data quality is a foundational concern in data-driven systems and is commonly defined across dimensions such as accuracy, completeness, consistency, timeliness, validity, and uniqueness[9]. Poor data quality directly affects downstream analytics and machine learning performance [10]. As a result, data quality management is treated as a continuous process spanning data ingestion, storage, transformation, and analysis [11]. Data cleaning and wrangling involve detecting and correcting errors such as missing values, duplicates, inconsistencies, and invalid formats. Traditional approaches model data cleaning as constraint enforcement or rule-based repair over structured dependencies. Modern techniques incorporate statistical and machine learning methods to detect anomalies and suggest repairs at scale [12]. A widely recognized foundation for data preprocessing is the “tidy data” framework [13], which formalizes principles for structuring datasets to facilitate analysis and transformation. Preprocessing steps such as normalization, encoding, and feature engineering remain critical, often having as much impact on model performance as the choice of learning algorithm itself [14].

Spreadsheet-based tools such as Microsoft Excel are still widely used for data manipulation, especially in business environments. However, their limitations in scalability, reproducibility, and complex transformation logic have been widely documented [15]. To address these limitations, tools like Tableau and Power BI provide low-code interfaces for data transformation and visualization, though they still require structured user interaction rather than natural language control [16]. Large language models (LLMs) such as GPT-3 and GPT-4 have demonstrated strong capabilities in reasoning, code generation, and instruction following. These models are increasingly applied to data management tasks, including SQL generation, data transformation, and explanation of data anomalies[17] . However, challenges such as hallucination, schema misinterpretation, and lack of grounding remain significant concerns.

To mitigate these issues, retrieval-augmented generation (RAG) approaches have been proposed, allowing models to access external knowledge sources during reasoning [18]. This improves factual consistency and enables better handling of domain-specific datasets. Conversational interfaces for databases and analytics allow users to express queries in natural language instead of SQL or programming languages. Early systems relied on rule-based semantic parsers, while modern systems use deep learning models trained on large datasets such as Spider for text-to-SQL tasks [19]. Recent surveys show that LLM-based approaches now dominate this field, though robustness and ambiguity resolution remain active research challenges [20]. Classic data cleaning research has also highlighted the importance of scalable detection and repair of inconsistencies, often using rule-based systems or statistical inference methods [21]. Despite progress, many of these systems still require expert intervention. Entity resolution and multi-table data integration remain challenging problems in data management, typically addressed using probabilistic matching and machine learning techniques [21]. These tasks are especially relevant in real-world datasets where inconsistencies span multiple tables and sources.

Across this body of work, a clear gap remains: most systems focus either on querying (text-to-SQL) or visualization, while relatively few support end-to-end conversational data manipulation, including cleaning, transformation, and restructuring of datasets in an intuitive manner. This gap motivates the development of systems like Quiksight, which aim to unify natural language interaction with practical data engineering operations.

3. Materials and Methods

3.1. Research Design

This work employs an applied research and system development design to create a practical solution for data accessibility. The research follows a development-oriented approach where the system is iteratively designed, tested, and evaluated based on functional performance. An Agile Incremental methodology was selected to facilitate continuous improvement. Development proceeded through distinct cycles, beginning with core file upload functionality and expanding to include conversational querying, automated manipulation, and data export capabilities.

3.2. System Development Methodology

The development process utilized an Agile Incremental model, ensuring modular testing and refinement. The initial increment established the Minimum Viable Product (MVP), enabling basic dataset interaction. Subsequent iterations integrated advanced features such as dataset profiling, natural language processing (NLP) for data cleaning, and transformation commands. This approach ensured that individual components were validated before final system integration.

3.3. System Design and Architecture

The architecture of Quiksight was designed using a modular approach to separate user interaction, data processing, and artificial intelligence components. The system follows a three-layer architecture consisting of the presentation layer, application layer, and LLM processing layer. The overall architecture of the proposed system is presented in Figure 1.

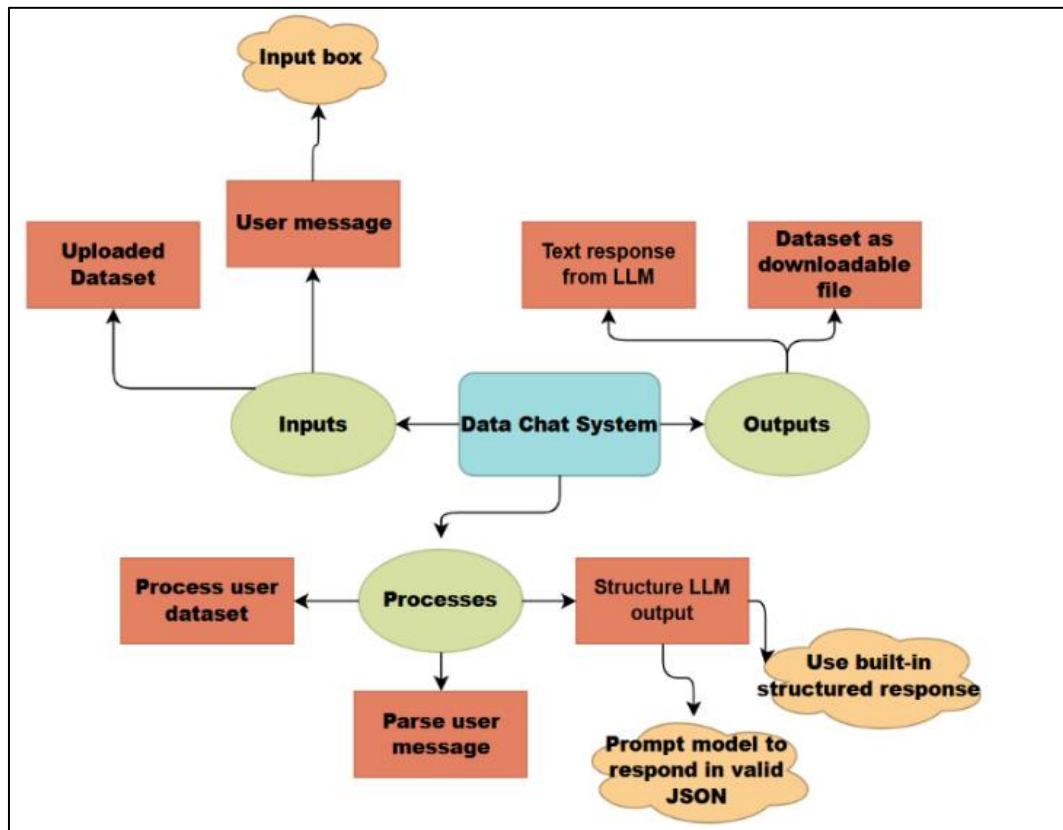


Figure 1 Conceptual Architecture of the Quiksight System

The presentation layer represents the user interface where users interact with the system. It provides functionalities for uploading datasets, submitting natural language queries, viewing processed results, and downloading generated outputs. This layer was implemented using HTML, JavaScript, and Tailwind CSS to provide a responsive and lightweight interface. The application layer serves as the central processing component of the system. It is responsible for receiving user requests, validating uploaded files, processing datasets, executing data manipulation operations, and managing communication between the user interface and the AI model. The backend was implemented using FastAPI, while

Pandas and NumPy were used for data processing tasks. The LLM layer provides the intelligence required for conversational interaction. Google Gemini was integrated through the OpenRouter API to interpret user instructions and convert natural language requests into executable data operations. The generated operations are processed by the backend and applied to the uploaded dataset before results are returned to the user.

3.4. System Workflow Design

The interaction process between the user and the system is illustrated using the sequence diagram shown in Figure 2.

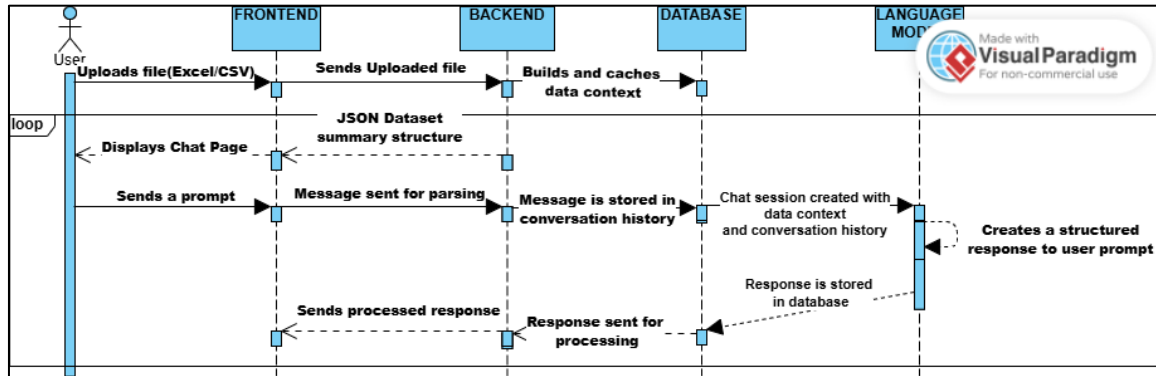


Figure 2 Sequence Diagram of the Quiksight System Workflow

The workflow begins when the user uploads a dataset through the web interface. The backend receives the file, validates the format, and converts the dataset into a structured DataFrame. The system then performs initial data profiling by identifying column names, data types, missing values, and basic statistical information. When a user submits a natural language query, the request is forwarded from the frontend to the backend. The backend combines the user query with dataset context and sends it to the LLM. The LLM interprets the request and generates an appropriate response or data manipulation instruction. The backend executes the required operation and returns the processed result to the user interface. The final stage involves exporting the processed dataset. Users can request the modified dataset in CSV or Excel format, which is generated by the backend and made available for download.

3.5. Functional Modelling

The functional behaviour of the system is represented using a use case diagram shown in Figure 3.

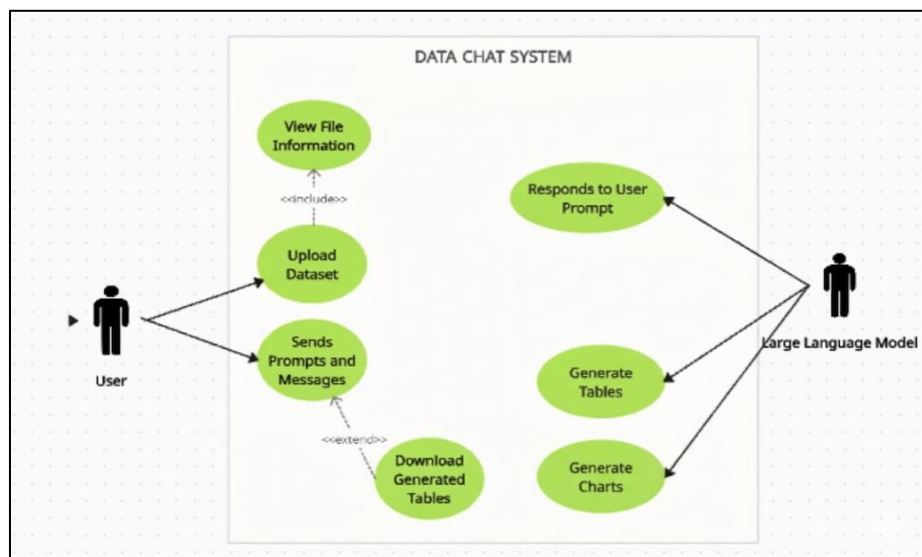


Figure 3 Use Case Diagram of Quiksight

The primary actor in the system is the user. The user interacts with the system through four major functions:

- Dataset Upload: Users upload CSV or Excel files which are validated and processed by the system.
- Dataset Preview: The system displays dataset information including column names, sample records, data types, and detected data quality issues.
- Conversational Data Manipulation: Users submit natural language instructions to perform operations such as filtering records, sorting values, renaming columns, summarizing information, and handling missing values.
- Dataset Export: Users can download the processed dataset for further use in external applications.

3.6. Implementation Tools and Technologies

The system was implemented using a combination of programming languages, frameworks, libraries, and artificial intelligence services. Python was selected as the primary backend programming language due to its strong ecosystem for data analysis and machine learning applications. Pandas was used for dataset manipulation, while NumPy supported numerical operations. FastAPI was used as the backend framework because of its high performance, lightweight structure, and support for building REST APIs. The frontend interface was developed using HTML, JavaScript, and Tailwind CSS. Google Gemini API was integrated as the Large Language Model responsible for natural language understanding and query interpretation. It was selected due to its ability to reason over structured data and generate relevant responses for conversational analysis. Additional development tools included Git and GitHub for version control, Visual Studio Code as the development environment, and Postman for API testing.

3.7. System Implementation Process

The implementation process consists of four main stages:

- Users upload datasets through the frontend interface. The backend validates the file extension and ensures that only supported CSV and Excel formats are accepted. Valid datasets are converted into Pandas DataFrames for processing.
- After successful upload, the system analyses the dataset to identify important characteristics such as column names, data types, missing values, and duplicate records. This information is used as contextual information for the LLM during subsequent interactions.
- The conversational analysis component allows users to interact with datasets using normal language. The user query and dataset information are provided to the LLM, which determines the required operation. The backend executes the generated instruction and returns the result. After processing, users can export the updated dataset. The system converts the DataFrame into CSV or Excel format and provides the file for download.

3.8. System Evaluation

The developed system was evaluated based on functionality, performance, and usability. Functional testing was conducted to verify that all major components operated correctly, including dataset upload, data manipulation, query processing, and file export. Performance evaluation focused on measuring query response time, dataset upload time, and processing speed across different dataset sizes. The accuracy of natural language interpretation was tested using multiple user queries involving different data manipulation tasks. Usability evaluation assessed the ease of interaction, interface simplicity, and effectiveness of the conversational approach for users with limited programming experience.

4. Results and Discussion

This section presents the implementation and evaluation of Quiksight, a conversational data analysis system designed to simplify dataset exploration and manipulation for non-technical users. The system integrates Large Language Model (LLM) capabilities with traditional data processing techniques to provide a natural language interface for structured data analysis.

The system follows a three-tier architecture consisting of a frontend layer, application layer, and data processing layer. The frontend was implemented using HTML, Tailwind CSS, and JavaScript to provide an interactive interface for dataset upload and conversational interaction. The backend was developed using FastAPI and Python, with Pandas used for data processing and Google Gemini integrated through OpenRouter for natural language understanding. Redis was used for maintaining session state and conversation context.

Users can upload CSV and Excel files through the interface, after which the system automatically processes the dataset and generates an initial summary. The uploaded file is converted into an in-memory DataFrame, allowing the system to

extract information such as dataset size, column names, data types, and missing value statistics. The upload workflow and generated dataset preview are shown in Figure 4, while the column-level statistical summary is presented in Figure 5

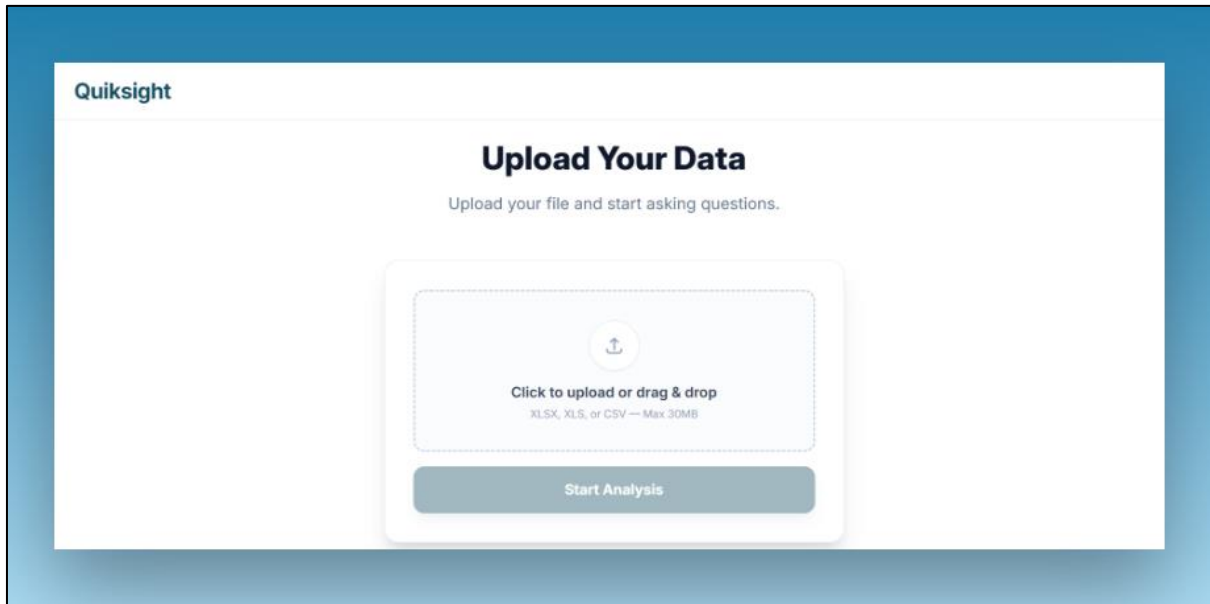


Figure 4 File Upload and Data Preview Interface

CUSTOMER_ID	TRANSACTION_COUNT	PROMO_USAGE_COUNT	QUANTITY	AVG_ORDER_VALUE	FIRST_PURCHASE_DATE	LAST_PURCHASE_DATE
21644	1	1	4	58.543	1/1/2021 1:14	1/1/2021 1:14
73192	1	2	2	37.015	1/1/2021 9:21	1/1/2021 9:21
9779	1	0	1	4.094	1/1/2021 16:04	1/1/2021 16:04
98867	1	0	1	14.923	1/2/2021 1:37	1/2/2021 1:37
48935	1	0	2	17.514	1/2/2021 9:29	1/2/2021 9:29

Figure 5 Column Statistics Preview

The conversational interface enables users to interact with datasets using natural language instructions. User queries are interpreted by the LLM and converted into executable Pandas operations that modify the active dataset. The processed results are returned as updated previews or analytical summaries without exposing implementation details or requiring programming knowledge. Figure 4 demonstrates the interaction between the user and the system through a natural language query, while Figure 5 illustrates the generation of data analysis outputs through the conversational interface.

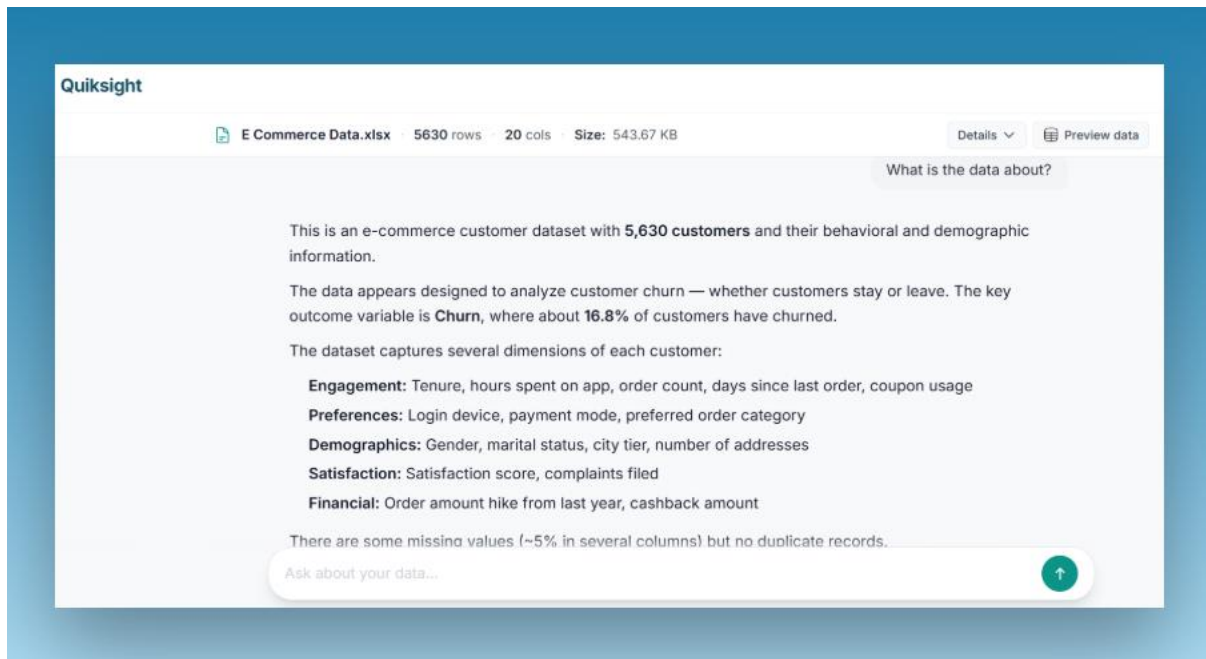


Figure 6 Chat Interface



Figure 7 Chat Interface Showing Data Visualization Result

The system was evaluated using datasets ranging from small files to medium-sized datasets up to 30MB. Testing covered common data manipulation operations, including filtering, sorting, column modification, missing value handling, grouping, and data export. The evaluation showed that Quiksight successfully executed the tested operations with reliable results and acceptable response times for the intended use case.

A comparative evaluation with existing conversational analytics platforms, including Julius.ai and PowerDrill.ai, showed that Quiksight emphasizes simplicity and accessibility. Unlike systems that expose programming workflows or focus on advanced analytics, Quiksight abstracts code execution and provides direct interaction with processed datasets. This design choice improves usability for users without programming backgrounds.

Although the system performs effectively on small and medium datasets, limitations were observed when processing larger files due to in-memory computation and external LLM API constraints. Additionally, the current version supports

only single-sheet Excel files. Future improvements will focus on enhanced dataset scalability, broader file format support, improved visualization capabilities, and optional local LLM deployment.

5. Conclusion

This study addressed the limited accessibility of data analysis tools for non-technical users by developing Quiksight, a web-based conversational data analysis system powered by a Large Language Model. The system allows users to upload CSV and Excel files and perform data manipulation tasks such as filtering, sorting, cleaning, and summarization through natural language commands. It was implemented using an Agile-Incremental approach with a three-tier architecture consisting of a frontend built with HTML, Tailwind CSS, and JavaScript, a FastAPI backend with Pandas for data processing, and Google Gemini for language understanding, with Redis used for session management.

The results show that all objectives were achieved, including dataset upload and preview, natural language-driven data manipulation, and system evaluation against existing tools such as Julius.ai and PowerDrill.ai. Experimental testing demonstrated that the system accurately interprets user commands and performs reliably on small to medium-sized datasets with acceptable response times.

Overall, the work demonstrates that Large Language Models can effectively enable intuitive, code-free data manipulation within lightweight web applications. Quiksight highlights the potential of conversational interfaces to bridge the gap between complex data processing systems and non-technical users by translating natural language directly into executable data transformations. The work contributes to the growing field of conversational AI for data management and provides a foundation for further research into scalable, user-friendly data analytics systems.

Compliance with ethical standards

Disclosure of conflict of interest

The Authors declare no competing or conflicting Interest.

References

- [1] A. Al Maruf, R. Paul, ; Mohammad, H. Imam, and ; Zahir Babar, "A SYSTEMATIC REVIEW OF THE ROLE OF SQL AND EXCEL IN DATA-DRIVEN BUSINESS DECISION-MAKING FOR ASPIRING ANALYSTS," *American Journal of Scholarly Research and Innovation*, vol. 1, no. 01, pp. 249–269, Dec. 2022, doi: 10.63125/N142CG62.
- [2] J. Leskovec, A. Rajaraman, J. David, and U. Frontmatter, "Mining of Massive Datasets Third Edition", Accessed: Jun. 17, 2026. [Online]. Available: www.cambridge.org
- [3] "DataViz: A Business Data Visualization System Using LLMs," 2025.
- [4] C. Qian *et al.*, "LLM Adoption in Data Curation Workflows: Industry Practices and Insights," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 1, Apr. 2025, doi: 10.1145/3706599.3719677;CSUBTYPE:STRING:CONFERENCE.
- [5] J. Leskovec, A. Rajaraman, J. David, and U. Frontmatter, "Mining of Massive Datasets Third Edition", Accessed: Jun. 17, 2026. [Online]. Available: www.cambridge.org
- [6] J. Gao *et al.*, "Efficiency with Rigor! A Trustworthy LLM-powered Workflow for Qualitative Data Analysis," *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*, vol. 1, Jan. 2025,
- [7] C. Qian *et al.*, "The Evolution of LLM Adoption in Industry Data Curation Practices," pp. 2024–2036, Dec. 2024, Accessed: Jun. 17, 2026. [Online]. Available: <https://arxiv.org/pdf/2412.16089>
- [8] L. Xie, C. Zheng, H. Xia, H. Qu, and C. Zhu-Tian, "WaitGPT: Monitoring and Steering Conversational LLM Agent in Data Analysis with On-the-Fly Code Visualization," *UIST 2024 - Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, vol. 1, Oct. 2024, doi: 10.1145/3654777.3676374;PAGEGROUP:STRING:PUBLICATION.
- [9] V. Anikó and V. Anikó, "Perspective Chapter: The Role of Data Quality Dimensions in Data-Driven Decision-Making," *Data Quality Matters - Best Practices for Integrity and Assurance*, Feb. 2026, doi: 10.5772/INTECHOPEN.1013233.

- [10] S. Mohammed *et al.*, “The Effects of Data Quality on Machine Learning Performance on Tabular Data,” vol. 1, Accessed: Jun. 17, 2026. [Online]. Available: <https://hai.stanford.edu/events/data-centric-ai-virtual-workshop>
- [11] I. Taleb, M. A. Serhani, C. Bouhaddioui, and R. Dssouli, “Big data quality framework: a holistic approach to continuous quality management”, doi: 10.1186/s40537-021-00468-0.
- [12] S. Krishnan, M. J. Franklin, K. Goldberg, J. Wang, and E. Wu, “ActiveClean: An Interactive Data Cleaning Framework For Modern Machine Learning”, doi: 10.1145/2882903.2899409.
- [13] H. Wickham, “Tidy Data,” *J. Stat. Softw.*, vol. 59, no. 10, pp. 1–23, Sep. 2014, doi: 10.18637/JSS.V059.I10.
- [14] E. Wang, D. Cook, and R. J. Hyndman, “A New Tidy Data Structure to Support Exploration and Modeling of Temporal Data,” *Journal of Computational and Graphical Statistics*, vol. 29, no. 3, pp. 466–478, Jul. 2020, doi: 10.1080/10618600.2019.1695624.
- [15] R. A. Gonçalves, “MICROSOFT EXCEL AND CONCEPTUAL SILENCING: CHALLENGES FOR STATISTICAL LITERACY IN THE DIGITAL AGE,” *Editora Impacto Científico*, pp. 107–127, Sep. 2025, doi: 10.56238/EDIMPACTO2025.034-007.
- [16] P. Charan, K. H. Latha, G. Neelamegam, R. Verma, S. Pal, and G. R. Teltumbade, “Transforming Big Data Challenges Into Opportunities: An In-Depth Analysis of Microsoft Power BI for Analytics and Visualization,” *Proceedings of the 2024 International Conference on Artificial Intelligence and Emerging Technology, Global AI Summit 2024*, pp. 1248–1252, 2024, doi: 10.1109/GLOBALAISUMMIT62156.2024.10947811.
- [17] JiangJuyong, WangFan, ShenJiasi, KimSungju, and KimSunghun, “A Survey on Large Language Models for Code Generation,” *ACM Transactions on Software Engineering and Methodology*, vol. 35, no. 2, pp. 1–72, Jan. 2026, doi: 10.1145/3747588.
- [18] M. Cheng *et al.*, “A Survey on Knowledge-Oriented Retrieval-Augmented Generation,” vol. 1, Mar. 2025, doi: XXXXXXXX.XXXXXXX.
- [19] T. Yu *et al.*, “Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 3911–3921, 2018, doi: 10.18653/V1/D18-1425.
- [20] F. El Boujddaini, A. Laguidi, and Y. Mejdoub, “A Survey on Text-to-SQL Parsing: From Rule-Based Foundations to Large Language Models,” *Lecture Notes in Networks and Systems*, vol. 1123 LNNS, pp. 266–272, 2024, doi: 10.1007/978-3-031-70411-6_41/SAVE-RESEARCH.
- [21] J. Hađina, J. Fogarty, and B. Jukić, “Innovative Data Modeling Concepts for Big Data Analytics: Probabilistic Cardinality and Replicability Notations,” *Array*, vol. 28, p. 100608, Dec. 2025, doi: 10.1016/J.ARRAY.2025.100608