AI-Powered SQL Database Chatbot

Using AI and NLP

Subikson M, Prem Kumar A, Zameer Ahmed F
Department of Computer Science and Engineering, Francis Xavier Engineering College Tirunelveli, Tamil Nadu, India
subiksonsubikson1@gmail.com, premk55662@gmail.com , zameerahmedf1@gmail.com

***Abstract* —** The growth of data in organizations has increased the need for smart systems that make it easy for non-technical users to interact with databases. Traditional database querying requires knowledge of Structured Query Language (SQL), which can be difficult for beginners and non-programmers.

This paper presents an AI-powered Multi-Database SQL Chatbot that allows users to interact with databases using natural language queries and voice commands. The proposed system combines Large Language Models (LLMs), speech recognition, text-to-speech functionality, and database management technologies to create an intelligent and user-friendly platform.

The chatbot supports both MySQL and SQLite databases and automatically converts user questions into executable SQL queries. The system also provides query execution results, schema visualization, and voice-based communication.

Using this solution improves accessibility, reduces query complexity, and enhances the overall user experience in database management systems.

Keywords— Artificial Intelligence, SQL Chatbot, Natural Language Processing, Voice Assistant, Database Management, Streamlit, LangChain, LLM.

I. INTRODUCTION

In recent years, Artificial Intelligence and Natural Language Processing have changed the way people use software applications. Many organizations such as schools, hospitals, businesses, and industries use databases to store large amounts of information. However, interacting with databases can be difficult for users who do not know how to write SQL commands.

To solve this problem, chatbot systems based on Artificial Intelligence have become an effective solution. These chatbots allow users to ask questions in natural language instead of writing SQL queries manually. The chatbot understands the user's request, generates the required SQL query, executes it on the database, and displays the results in a simple and understandable format.

This paper presents an AI-powered Multi-Database SQL Chatbot that supports both text and voice communication. The system uses Large Language Models with LangChain to generate SQL queries dynamically. The chatbot can also understand voice commands and respond through voice output. The system supports databases such as MySQL and SQLite and can automatically identify tables and schema information.

The main goal of this project is to make database interaction easier, faster, and more accessible for all users. Artificial Intelligence and Natural Language Processing help users communicate with databases without requiring SQL knowledge.

II. EXISTING SYSTEM

Current database management systems mainly depend on users writing SQL queries manually. To retrieve information from databases, users need knowledge of SQL syntax and database structure. Traditional systems usually provide command-line interfaces or complicated graphical tools that are difficult for beginners and non-technical users to understand.

In addition, many systems do not support voice interaction, making them less accessible for users who prefer speech-based communication.

Existing systems also face several problems:

- Dependence on SQL knowledge.
- Limited database compatibility.
- Incorrect query generation for complex questions.
- Lack of real-time schema analysis.
- Poor user interaction and accessibility.
- Limited voice communication support.

Because of these limitations, users often face difficulties while retrieving information from databases. These drawbacks created the need for an intelligent and user-friendly database interaction system.

III. PROPOSED SYSTEM

The AI-powered Multi-Database SQL Chatbot is designed to provide intelligent interaction between users and databases. The system can understand user queries through both text and voice input and uses Artificial Intelligence technologies to process them efficiently.

The proposed system contains several important modules.

User Interface Module

The system uses Streamlit to create an interactive web-based interface. Users can enter queries through text or voice commands.

ISSN: 3107-6513

Voice Processing Module

The SpeechRecognition module converts spoken input into text format, allowing users to communicate naturally with the chatbot.

Natural Language Processing Module

This module analyzes the user query and identifies the required database operation. LangChain and Groq LLM models are used to generate SQL queries dynamically.

SQL Query Generation Module

The generated SQL query is validated before execution. Duplicate and invalid queries are removed to improve system performance and reliability.

Database Connectivity Module

The chatbot supports multiple databases such as MySQL and SQLite. SQLAlchemy is used to establish secure database connections.

Result Visualization Module

The retrieved results are displayed in tabular format using Pandas DataFrames. The chatbot also summarizes the results in natural language.

Text-to-Speech Module

The chatbot converts important responses into speech using the pyttsx3 library, allowing users to hear the responses clearly.

IV. METHODOLOGY

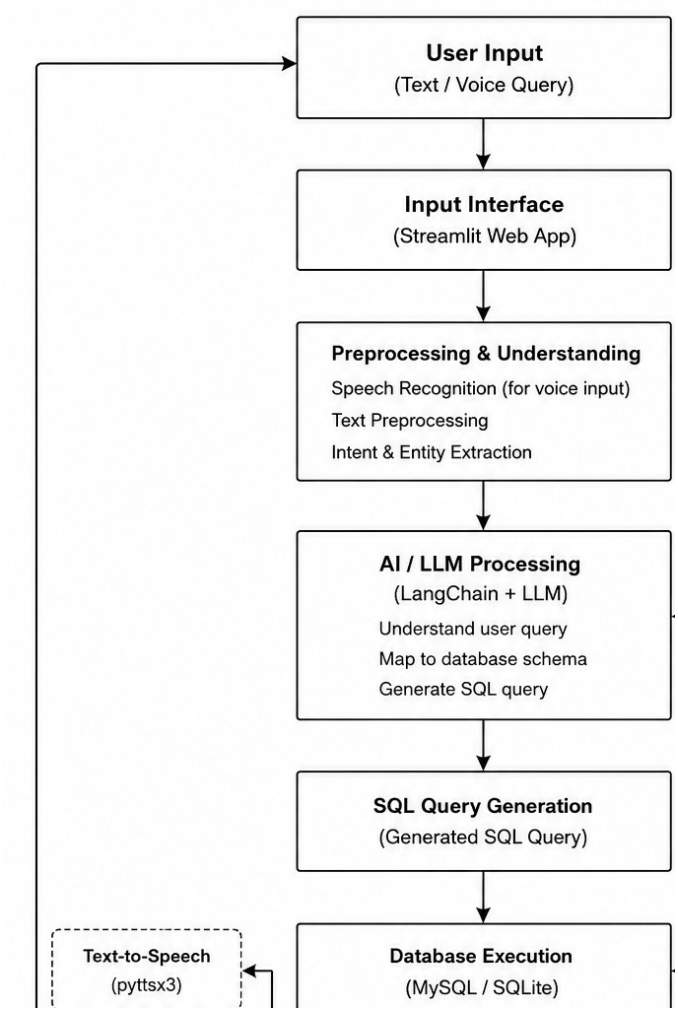
The AI-powered Multi-Database SQL Chatbot is designed to simplify database interaction using Artificial Intelligence, Natural Language Processing, and voice-based technologies. The system follows a structured workflow to process user queries accurately and efficiently.

The system first collects user queries through text input or voice commands. Users can type their questions directly into the chatbot interface or speak through a microphone. If voice input is used, the SpeechRecognition module converts speech into text format.

After receiving the input, the system uses Natural Language Processing techniques to analyze the query. The chatbot identifies important keywords, database entities, and user intentions. Based on this understanding, the Large Language Model generates the required SQL query automatically.

The generated SQL query is validated to remove duplicate or incorrect queries. After validation, the query is executed on the selected database. The system supports databases such as MySQL and SQLite.

The retrieved data is processed and displayed in tabular format. The chatbot also converts the database results into simple human-readable summaries.



The system further provides voice responses using text-to-speech technology. Conversation history is maintained to support continuous interaction between users and the chatbot.

The overall methodology combines Artificial Intelligence, voice assistance, and multi-database support to create an efficient and user-friendly database interaction system

V. FUTURE EXTENSION

Feature extraction is an important part of the AI-powered Multi-Database SQL Chatbot system. In this stage, the system identifies and extracts meaningful information from user queries to understand user intentions accurately.

The system processes both text and voice input. If voice input is provided, it is first converted into text format. Natural Language Processing techniques are then used to identify important keywords, table names, column names, database entities, and query conditions.

The chatbot extracts features such as selecting records, counting values, filtering data, searching for information, and retrieving complete table details. It also identifies important conditions such as names, dates, categories, and numerical values mentioned in the query.

These extracted features help the Large Language Model generate accurate SQL queries dynamically. Feature extraction

improves query understanding, reduces ambiguity, and increases the overall performance of the chatbot system.

VI. IMPLEMENTATION

The implementation phase focuses on building the AI-powered Multi-Database SQL Chatbot system using Artificial Intelligence, Natural Language Processing, and database technologies.

The system first establishes connections with databases such as MySQL and SQLite using SQLAlchemy. Database schema details, tables, and columns are identified so that the chatbot can understand database structures properly.

A user-friendly interface is developed using Streamlit, allowing users to interact with the system through text or voice input. When a user enters a query, the chatbot processes it using Natural Language Processing techniques.

If the query is provided through voice input, the SpeechRecognition module converts speech into text format. The processed query is analyzed to identify keywords, database entities, and user intentions.

The Large Language Model then generates the required SQL query automatically. The generated SQL statement is validated before execution to remove duplicate or incorrect queries.

After execution, the retrieved results are displayed in tabular format and summarized in natural language. The system also uses text-to-speech technology to provide voice responses to users.

The implementation successfully demonstrates how Artificial Intelligence can simplify database interaction for both technical and non-technical users.

VI. Security Alert and Prevention Measures

Security is an important aspect of the AI-powered Multi-Database SQL Chatbot system because the chatbot directly interacts with databases containing important information.

The system includes several security measures to protect databases from unauthorized access and malicious activities. User inputs are validated before processing to prevent harmful or invalid queries.

The chatbot validates SQL queries before execution to ensure that only safe and properly structured queries are allowed. Suspicious or malformed queries are automatically blocked to reduce the risk of SQL injection attacks and database misuse.

The system also secures access to Large Language Model services using API key validation. Database credentials are protected using secure configuration methods to prevent unauthorized access.

Additional error handling and exception management techniques are implemented to detect abnormal activities and system failures. These security measures improve the reliability, safety, and trustworthiness of the chatbot system.

VII FUTURE WORK

The AI-powered Multi-Database SQL Chatbot system can be improved further in the future using advanced Artificial Intelligence and Deep Learning techniques. These technologies can increase the accuracy of SQL query generation and improve database interaction.

More advanced Natural Language Processing models can be integrated to better understand complex user queries and improve conversational capabilities.

The system can also support additional databases such as PostgreSQL, MongoDB, and cloud-based database systems to provide greater compatibility and scalability.

Future versions may include advanced data visualization features such as dashboards, charts, and graphical reports for better understanding of database results. Multilingual support can also be added to allow users to communicate with the chatbot in different languages.

Additional security features, intelligent query optimization, and personalized user interaction can further improve the efficiency and usability of the chatbot system.

REFERENCES

[1] Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke, *An LLM-Driven Chatbot in Higher Education for Databases and Information Systems*, 2025.

[2] Atharv Deshmukh, Manali Gawade, and Ronit Fulari, AI-Powered SQL Assistant: Transforming Natural Language into Optimized SQL Queries, 2025.

[3] Girish R. Swasthik K. and Veera Mani Kandan, *AI Powered SQL Query Generator and Data Visualizer*, 2024.

[4] Arnav Jha, Naman Anand, and H. Karthikeyan, Conversion of Natural Language Text to SQL Queries Using Generative AI, 2025.

[5] Xiaodong Su, Yang Gu, Peng Wang, Wei Gu, Lincheng Qi, and Jingwei He, A Robust Natural Language Text-to-SQL Generation Framework with Dynamic Strategies Based on LLMs, 2026.

[6] Daniel Jurafsky and James H. Martin, *Speech and Language Processing*, Pearson Education, 2021.

[7] Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1997.

[8] Sudhan S., Prasanth P. Nair, and M. Thushara, *Text-to-Speech and Speech-to-Text Models: A Systematic Examination of Diverse Approaches*, in 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), IEEE, 2024, pp. 1–8.

[9] Yogesh Kumar Dwivedi, Nripendra Pandey, Wendy Currie, and Alin Micu, Leveraging ChatGPT and Other Generative Artificial Intelligence (AI)-Based Applications in Various Industries: Practices, Challenges and Research Agenda, International Journal of Contemporary Management, vol. 36, no. 1, pp. 1–12, 2024.

[10] Mike Bayer, SQLAlchemy: SQL Toolkit and Object Relational Mapper for Python Applications, SQLAlchemy Documentation, 2024.

ISSN: **3107-6513**

[11] LangChain Developers, LangChain Framework for AI-Powered Natural Language Applications and SQL Query Generation, 2024.

[12] Streamlit Inc., Streamlit Framework for Interactive Web-Based Data Applications, 2024.

[13] Groq Inc., Groq AI Large Language Model API and Intelligent Query Processing Framework, 2024.

[14] Anthony Zhang, SpeechRecognition Python Library for Voice-Based User Interaction Systems, 2024.

[15] David Jones, pyttsx3 Text-to-Speech Conversion Library for Intelligent Voice Assistant Applications, 2024.